SIEMENS

SIMATIC

PCS 7/APACS+ ES Library User's Guide for V6.1

Manual

Preface, Contents

Introduction 1

Library Function Blocks 2

Template Charts 3

Message Class 4

Differences between APACS+ and PCS 7 5

Customizing the Library 6

Peformance Data 7

Edition 04/2006
A5E00595270-01

Safety Guidelines

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring to property damage only have no safety alert symbol. The notices shown below are graded according to the degree of danger.

Danger
indicates that death or severe personal injury **will** result if proper precautions are not taken.

Warning
indicates that death or severe personal injury **may** result if proper precautions are not taken.

Caution
with a safety alert symbol indicates that minor personal injury can result if proper precautions are not taken.

Caution
without a safety alert symbol indicates that property damage can result if proper precautions are not taken.

Notice
indicates that an unintended result or situation can occur if the corresponding notice is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notices in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

Prescribed Usage

Note the following:

Warning
This device and its components may only be used for the applications described in the catalog or the technical description, and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens.
Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG.
The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Automation and Drives
Postfach 4848
90437 NÜRNBERG
GERMANY

Copyright © Siemens AG 2005
A5E00595270-01

Siemens AG 2006
Technical data subject to change.

Preface

Purpose of the Manual

This manual provides information necessary to set up PCS 7/APACS+™ ES Library for V6.1.

Required Basic Knowledge

Readers are presumed to be knowledgeable in the use of PCS 7 and APACS+.

Where is this Manual valid?

This manual is valid for the software package PCS 7/APACS+ OS for V6.1.

Training Centers

Siemens Technical Training Center provides extensive training for all levels of plant personnel to ensure optimal performance from APACS+ and PCS 7 control systems. Classes include extensive hands-on activities using appropriate equipment, making the training directly and immediately applicable.

On-line information is available: http://www.sea.siemens.com/sitrain

Siemens also offers a number of training courses to familiarize you with the SIMATIC S7 automation system. Please contact your regional training center or our central training center in D 90327 Nuremberg, Germany for details:

Telephone: +49 (911) 895-3200.

Internet: http://www.sitrain.com

A&D Technical Support

Worldwide, available 24 hours a day:

United States: Johnson City, TN	**Worldwide:** Nürnberg	**Asia / Australia**: Beijing
Technical Support and Authorization Local time: Monday to Friday 8:00 AM to 5:00 PM Telephone:+1 (423) 262 2522 or +1 (800) 333-7421 (USA only) Fax:+1 (423) 262 2289 Mail to: techsupport.sea@siemens.com GMT: -5:00	**Technical Support** 24 hours a day, 365 days a year Phone:+49 (180) 5050-222 Fax:+49 (180) 5050-223 E-Mail: ad.support@siemens.com GMT:+1:00 **Authorization** Local time: Monday to Friday 8:00 AM to 5:00 PM Phone: +49 (180) 5050-222 Fax: +49 (180) 5050-223 Mail to: ad.support@siemens.com GMT: +1:00	**Technical Support and Authorization** Local time: Monday to Friday 8:00 AM to 5:00 PM Phone:+86 10 64 75 75 75 Fax:+86 10 64 74 74 74 Mail to:ad.support.asia@siemens.com GMT:+8:00
Automation and Drives Service and Support International http://www.siemens.com/automation/service&support		
The languages of the SIMATIC Hotlines and the authorization hotline are generally German and English.		

1 PCS 7 ES Option for APACS+ 1-1

1.1 Introduction - The PCS 7 ES Option for APACS+™ ... 1-1
1.2 Library Description ... 1-2

2 Library Function Blocks 2-1

2.1 Introduction ... 2-1
2.2 AALM_AFB ... 2-3
2.3 AM_AFB ... 2-14
2.4 ANALOG_ALM_AFB ... 2-17
2.5 BLK_VLV_1OUT_AFB ... 2-24
2.5.1 BLK_VLV_1OUT: Block Valve with One ... 2-24
2.6 BLK_VLV_2OUT_AFB ... 2-29
2.7 CASCADE_AFB ... 2-34
2.8 DALM_AFB ... 2-55
2.9 DISCRETE_ALM_AFB ... 2-61
2.10 EXT_SET_AFB ... 2-64
2.11 MOTOR_A_AFB ... 2-76
2.12 MOTOR_A_ALM_AFB ... 2-85
2.13 MOTOR_1OUT_AFB ... 2-92
2.14 MOTOR_2OUT_AFB ... 2-97
2.15 PID_AFB ... 2-102
2.16 PRIMARY_AFB ... 2-105
2.17 RATIO_SET_AFB ... 2-117
2.18 SECONDARY_AFB ... 2-130
2.19 SETPNT_AFB ... 2-143
2.20 SINGLE_LOOP_AFB ... 2-148
2.21 SINGLE_LP_SS_AFB ... 2-158
2.22 VALVE_A_ALM_AFB ... 2-171
2.23 VALVE_A_AFB ... 2-178

3 Template Charts 3-1

3.1 ANALOG_ALM_AT ... 3-2
3.2 BLK_VLV_1OUT_AT ... 3-3
3.3 BLK_VLV_2OUT_AT ... 3-4
3.4 CASCADE_AT ... 3-5
3.5 DISCRETE_ALM_AT ... 3-7
3.6 EXT_SET_AT ... 3-8
3.7 MOTOR_A_ALM_AT ... 3-10
3.8 MOTOR_1OUT_AT ... 3-11
3.9 MOTOR_2OUT_AT ... 3-12
3.10 PRIMARY_SECONDARY_AT ... 3-13
3.11 RATIO_SET_AT ... 3-15
3.12 SINGLE_LOOP_AT ... 3-17
3.13 SINGLE_LP_SS_AT ... 3-19
3.14 VALVE_A_ALM_AT ... 3-21

4 Message class 4-1

5 Differences between APACS+ and PCS 7 5-1

5.1 General Differences ... 5-1
5.2 Alarm Differences ... 5-5
5.3 APACS+ versus PCS 7 Controller Algorithm Comparison ... 5-6

6 Customizing the Library 6-1

7 Performance Data 7-1

1 PCS 7 ES Option for APACS+

1.1 Introduction - The PCS 7 ES Option for APACS+™

The PCS 7 ES Option for APACS+ is a SIMATIC® PCS 7 library intended for users who have an APACS+ background. The library also contains blocks from the APACS+ Basic Application Library that have been ported to both PCS 7 FBs and PCS 7 Continuous Function Charts (CFCs).

1.2 Library Description

The PCS 7 ES Option for APACS+ is a SIMATIC® PCS 7 library is composed of Source Charts (CFCs) and Structured Control Language (SCL) source files that emulate many of the Derived Function Blocks (DFBs) in the APACS+ Basic Application Library. The source charts and source files are also provided as compiled APACS+ function block types that may be used with their matching faceplates and graphic symbols. Template Charts show how the APACS+ function block types are typically used.

This library supplies a set of building blocks for users who want to construct PCS 7 applications based on APACS+ structures. These components are:

Ported APACS+ Blocks	These function blocks provide the functionality of APACS+ standard function blocks.
Source (CFC) Charts and Source (SCL) Files	These charts and files provide the functionality of APACS+ derived function blocks and give users the ability to customize the types and create new types.
APACS+ Function Block Types	These function blocks provide the functionality of APACS+ derived function blocks in a form that has connected faceplates and graphic symbols.
Template (CFC) Charts	These charts provide examples of APACS+ function block types integrated with other PCS 7 blocks to facilitate easier user implementation within the PCS 7 ES.

Additional Topics:

Faceplates and Graphic Symbols	These pre-configured Human-Machine Interface (HMI) elements provide the ability to display/modify data associated with an APACS+ function block type.
Customizing the Library	These instructions describe customization of the source charts and files, including compilation to create new block types and modification of the graphical elements.

2 Library Function Blocks

2.1 Introduction

A PCS 7 function block serves the same purpose as a function block in APACS+. A function block runs a pre-configured algorithm that is not user configurable. There are block parameters that can be used to define the operation of the algorithm, just as softlist parameters were used in APACS+. Block parameters can be displayed as input/output nubs on the block by the user.

PCS 7 Function Block	**PCS 7 FB Number**	**Original APACS+ Standard Function Block**
AM_AFB	FB 500	AM (Auto/Manual)
PID_AFB	FB 501	PID (Proportional-Integral-Derivative)
SETPNT_AFB	FB 502	SETPNT (Setpoint)
AALM_AFB	FB 503	HLLAALM (Hi-Level Link Analog Alarm)
DALM_AFB	FB 504	HLLDALM (Hi-Level Link Discrete Alarm)
MOTOR_A_AFB	FB 505	MOTOR_A (Standard Motor)
VALVE_A_AFB	FB 506	VALVE_A (Standard Valve)

A PCS 7 chart is used in the creation of a Function Block Type. In APACS+ a Derived Function Block (DFB or UDFB) is equivalent to a Function Block Type. PCS 7 faceplates can be connected to a single block in the controller. The CFC for Single Loop contains four blocks (Setpoint, PID, AutoManual and Alarm). Therefore, the Single Loop faceplate could not be connected to the CFC. In essence, the Function Block Type is a compiled version of the CFC. We have provided the CFC to make it easier to customize the function blocks provided. The user can modify the CFC chart and then compile it to create a new Function Block Type.

The blocks marked by an * below are implemented in the SCL language rather than via the CFC charts. The user may also customize these Function Block Types by using the SCL Editor in SIMATIC Manager to modify and re-compile them.

	PCS 7 Function Block	**PCS 7 FB Number**	**Original APACS+ Basic Application Library Block**
	ANALOG_ALM_AFB	FB 600	ANALOG_ALARM (Analog Alarm)
*	BLK_VLV_1OUT_AFB	FB 601	BLK_VLV_1OUT (Block Valve with One Output)
*	BLK_VLV_2OUT_AFB	FB 602	BLK_VLV_2OUT (Block Valve with Two Outputs)
	CASCADE_AFB	FB 603	CASCADE (Cascade Controller)
	DISCRETE_ALM_AFB	FB 604	DISCRETE_ALARM (Discrete Alarm)
	EXT_SET_AFB	FB 605	EXT_SET (External Setpoint Controller)
*	MOTOR_1OUT_AFB	FB 606	MOTOR_1OUT (Motor with One Output)
*	MOTOR_2OUT_AFB	FB 607	MOTOR_2OUT (Motor with Two Outputs)
	PRIMARY_AFB	FB 608	PRIMARY (Primary Controller)
	RATIO_SET_AFB	FB 609	RATIO_SET (Ratio Setpoint Controller)
	SECONDARY_AFB	FB 610	SECONDARY (Secondary Controller)
	SINGLE_LOOP_AFB	FB 611	SINGLE_LOOP (Single Loop Controller)
	SINGLE_LP_SS_AFB	FB 612	SINGLE_LOOP_SS (Single Loop Controller w/ Standby Sync)
	MOTOR_A_ALM_AFB	FB 613	MOTOR_A_ALM (Motor with Alarm)
	VALVE_A_ALM_AFB	FB 614	VALVE_A_ALM (Valve with Alarm)

2.2 AALM_AFB

2.2.1 AALM_AFB: Analog Alarm

Function Block Name	AALM_AFB
Function Block Number	FB503

Function

The Analog Alarm Function Block (AALM_AFB) is intended for detecting and signaling abnormal data values. This block contains an Enable input, Alarm output, and NAK (not acknowledged) output for each of the following alarm types:

• High • Low • Absolute Deviation • High Deviation • Low Deviation • High High • Low Low	• Out Of Range • Quality Not Good • Absolute Rate Of Change • Increasing Rate Of Change • Decreasing Rate Of Change

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32).

Start-up Characteristics

This function block has no special startup properties.

Operator Control and Monitoring (M&C Relevant)

There is no associated faceplate for this block.

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

ALARM_8P ID	Message #	Message Class	Default Message Text
EV_ID	1	Warning - above	HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID	2	Warning - below	LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID	3	Alarm - above	HIGH_DEV - $$BlockComment$$ - Limit = @6%f@
EV_ID	4	Alarm - below	LOW_DEV - $$BlockComment$$ - Limit = @7%f@
EV_ID	5	Alarm - above	ABSOLUTE_DEV - $$BlockComment$$ - Limit = @8%f@
EV_ID	6	PLC Process Control Message - Failure	OUT_OF_RANGE - $$BlockComment$$
EV_ID	7	PLC Process Control Message - Failure	QUALITY_NOT_GOOD - $$BlockComment$$
EV_ID2	1	Alarm - above	HIGH_ HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID2	2	Alarm - below	LOW_ LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID2	3	Warning - above	INC_ROC - $$BlockComment$$ - Limit = @6%f@
EV_ID2	4	Warning - below	DEC_ROC - $$BlockComment$$ - Limit = @7%f@
EV_ID2	5	Warning - above	ABS_ROC - $$BlockComment$$ - Limit = @8%f@

Assignment of auxiliary process values to the block parameters

ALARM_8P ID	Value	Block Parameter
EV_ID	@1%s@	BA_NA
EV_ID	@2%d@	BA_ID
EV_ID	@3%d@	STEP_NO
EV_ID	@4%f@	HIGH_LIMIT
EV_ID	@5%f@	LOW_LIMIT
EV_ID	@6%f@	H_DEV_LIMIT
EV_ID	@7%f@	L_DEV_LIMIT
EV_ID	@8%f@	A_DEV_LIMIT
EV_ID2	@1%s@	BA_NA
EV_ID2	@2%d@	BA_ID
EV_ID2	@3%d@	STEP_NO
EV_ID2	@4%f@	HH_LIMIT
EV_ID2	@5%f@	LL_LIMIT
EV_ID2	@6%f@	I_ROC_LIMIT
EV_ID2	@7%f@	D_ROC_LIMIT
EV_ID2	@8%f@	A_ROC_LIMIT

2.2.1.1 AALM_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
PV	REAL	Process Value – This input is the value to be monitored for alarms.	0	True	
DEV	REAL	Deviation – The difference between this input and the (PV) input is monitored when a deviation alarm is selected.	0	False	
EN_HIGH	BOOL	When TRUE, the High alarm is enabled.	False	True	
EN_LOW	BOOL	When TRUE, the Low alarm is enabled.	False	True	
EN_H_DEV	BOOL	When TRUE, the High Deviation alarm is enabled.	False	True	
EN_L_DEV	BOOL	When TRUE, the Low Deviation alarm is enabled.	False	True	
EN_A_DEV	BOOL	When TRUE, the Absolute Deviation alarm is enabled.	False	True	
EN_OORANGE	BOOL	When TRUE, the Out of Range alarm is enabled	False	True	
EN_QUALITY	BOOL	When TRUE, the Quality Not Good alarm is enabled.	False	True	
EN_HH	BOOL	When TRUE, the High High alarm is enabled.	False	True	
EN_LL	BOOL	When TRUE, the Low Low alarm is enabled.	False	True	
EN_I_ROC	BOOL	When TRUE, the Increasing Rate of Change alarm is enabled.	False	True	
EN_D_ROC	BOOL	When TRUE, the Decreasing Rate of Change alarm is enabled.	False	True	
EN_A_ROC	BOOL	When TRUE, the Absolute Rate of Change alarm is enabled.	False	True	
OOS	BOOL	Out of Service indicator.	False	True	
QBAD	BOOL	Bad quality indicator – This value is a hardware diagnostic showing whether the quality of (PV) is bad. This input should be connected to the QBAD output of the channel driver feeding (PV). It is initialized to FALSE so that the quality is good if the parameter is unconnected.	False	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
EV_ID	DWORD	Event ID – This value is a unique message number supplied by PCS 7. There is one for each message (i.e. ALARM_8P) block called.	None	False	
EV_ID2	DWORD	Event ID - This value is a unique message number supplied by PCS 7. There is one for each message (i.e. ALARM_8P) block called.	None	False	
BA_NA	STRING[32]	Batch name	‘‘	False	
BA_ID	DWORD	Batch ID number	0	False	
STEP_NO	DWORD	Batch step number	0	False	

Note

This list differs from the inputs of the AALM_AFB block in APACS+ as follows:

- In APACS+, there are only four enables. There is now an enable for every alarm type.
- The parameter QBAD was added to get quality information from the control system.
- The parameters EV_ID and EV_ID2 were added to support the message system.
- Batch parameters BA_NA, BA_ID, STEP_NO are added to pass batch values in alarm messages.

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
ALM_HIGH	BOOL	When TRUE, the High alarm is detected and enabled.	False	False	
NAK_HIGH	BOOL	When TRUE, the High alarm is tripped but not acknowledged.	False	False	
ALM_LOW	BOOL	When TRUE, the Low alarm is detected and enabled.	False	False	
NAK_LOW	BOOL	When TRUE, the Low alarm is tripped but not acknowledged.	False	False	
ALM_H_DEV	BOOL	When TRUE, the High Deviation alarm is detected and enabled.	False	False	
NAK_H_DEV	BOOL	When TRUE, the High Deviation alarm is tripped but not acknowledged.	False	False	
ALM_L_DEV	BOOL	When TRUE, the Low Deviation alarm is detected and enabled.	False	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
NAK_L_DEV	BOOL	When TRUE, the Low Deviation alarm is tripped but not acknowledged.	False	False	
ALM_A_DEV	BOOL	When TRUE, the Absolute Deviation alarm is detected and enabled.	False	False	
NAK_A_DEV	BOOL	When TRUE, the Absolute Deviation alarm is tripped but not acknowledged.	False	False	
ALM_OORANGE	BOOL	When TRUE, the Out of Range alarm is detected and enabled.	False	False	
NAK_OORANGE	BOOL	When TRUE, the Out of Range alarm is tripped but not acknowledged.	False	False	
ALM_QUALITY	BOOL	When TRUE, Quality Not Good alarm is detected and enabled.	False	False	
NAK_QUALITY	BOOL	When TRUE, the Quality Not Good alarm is tripped but not acknowledged.	False	False	
ALM_HH	BOOL	When TRUE, High High alarm is detected and enabled.	False	False	
NAK_HH	BOOL	When TRUE, the High High alarm is tripped but not acknowledged.	False	False	
ALM_LL	BOOL	When TRUE, the Low Low alarm is detected and enabled.	False	False	
NAK_LL	BOOL	When TRUE, the Low Low alarm is tripped but not acknowledged.	False	False	
ALM_I_ROC	BOOL	When TRUE, the Increasing Rate of Change alarm is detected and enabled.	False	False	
NAK_I_ROC	BOOL	When TRUE, Increasing Rate of Change alarm is tripped but not acknowledged.	False	False	
ALM_D_ROC	BOOL	When TRUE, the Decreasing Rate of change alarm is detected and enabled.	False	False	
NAK_D_ROC	BOOL	When TRUE, the Decreasing Rate of change alarm is tripped but not acknowledged.	False	False	
ALM_A_ROC	BOOL	When TRUE, the Absolute Rate of Change alarm is detected and enabled.	False	False	
NAK_A_ROC	BOOL	When TRUE, Absolute Rate of Change alarm is tripped but not acknowledged.	False	False	
MSG1_ERR	BOOL	First ALARM_8P error	False	False	
MSG1_STAT	WORD	First ALARM_8P status	W#16#0	False	
MSG2_ERR	BOOL	Second ALARM_8P error	False	False	
MSG2_STAT	WORD	Second ALARM_8P status	W#16#0	False	

Note

This list differs from the outputs of the AALM_AFB block in APACS+ as follows:

- In APACS+, there are only four alarms and four NAKs. There is now one of each for every alarm type.
- The STATUS and CMD outputs have been removed.
- MSG1_ERR, MSG1_STAT, MSG2_ERR, and MSG2_STAT have been added to provide the status of the message processing. For specific values, see the online help for ALARM_8P.

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
HIGH_LIMIT	REAL	The value (in engineering units) to which the (PV) input is compared for the High alarm.	0	True	
HIGH_DEADBAND	REAL	The amount of engineering units by which (PV) must be within the limit for the High alarm to clear.	0.5	True	
HIGH_DELAYIN	TIME	The amount of time the input must reach or exceed the High limit to trip an alarm.	T#0d0h0m0s0ms	False	
HIGH_DELAYOUT	TIME	The amount of time the input must be within the High limit and deadband to cause the alarm to clear.	T#0d0h0m0s0ms	False	
LOW_LIMIT	REAL	The value (in engineering units) to which the (PV) input is compared for the Low alarm.	0	True	
LOW_DEADBAND	REAL	The amount of engineering units by which (PV) must be within the limit for the Low alarm to clear.	0.5	True	
LOW_DELAYIN	TIME	The amount of time the input must reach or exceed the Low limit to trip an alarm.	T#0d0h0m0s0ms	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
LOW_DELAYOUT	TIME	The amount of time the input must be within the Low limit and deadband to cause the alarm to clear.	T#0d0h0m0s0ms	False	
H_DEV_LIMIT	REAL	The value (in engineering units) to which (PV) – (DEV) is compared for the High Deviation alarm.	0	True	
H_DEV_DEADBAND	REAL	The amount of engineering units by which (PV) – (DEV) must be within the limit for the High Deviation alarm to clear.	0.5	True	
H_DEV _DELAYIN	TIME	The amount of time the input must reach or exceed the High Deviation limit to trip an alarm.	T#0d0h0m0s0ms	False	
H_DEV_DELAYOUT	TIME	High Deviation Alarm Delay Out input. It is the amount of time the input must be within the H_DEV_LIMIT value and H_DEV_DEADBAND value to cause the alarm to clear.	T#0d0h0m0s0ms	False	
L_DEV_LIMIT	REAL	The value (in engineering units) to which (DEV) – (PV) is compared for the Low Deviation alarm.	0	True	
L_DEV_DEADBAND	REAL	The amount of engineering units by which (DEV) – (PV) must be within the limit for the Low Deviation alarm to clear.	0.5	True	
L_DEV _DELAYIN	TIME	The amount of time the input must reach or exceed the Low Deviation limit to trip an alarm.	T#0d0h0m0s0ms	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
L_DEV_DELAYOUT	TIME	The amount of time the input must be within the Low Deviation limit and deadband to cause the alarm to clear.	T#0d0h0m0s0ms	False	
A_DEV_LIMIT	REAL	The value (in engineering units) to which ABS [(PV) – (DEV)] must be within the limit for the Absolute Deviation alarm to clear.	0	True	
A_DEV_DEADBAND	REAL	The amount of engineering units by which ABS [(PV) – (DEV)] must be within the limit for the Absolute Deviation alarm to clear.	0.5	True	
A_DEV _DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Deviation limit to trip an alarm.	T#0d0h0m0s0ms	True	
A_DEV_DELAYOUT	TIME	The amount of time the input must be within the Absolute Deviation limit and deadband to cause the alarm to clear.	T#0d0h0m0s0ms	False	
OORANGE_DELAYIN	TIME	The amount of time the input must be outside the limits EINH, EINL, to trip the Out of Range alarm.	T#0d0h0m0s0ms	False	
OORANGE_DELAYOUT	TIME	The amount of time the input must be within the limits EINH, EINL to cause the Out of Range alarm to clear.	T#0d0h0m0s0ms	False	
QUALITY_DELAYIN	TIME	The amount of time the input quality must be bad to trip the Quality Not Good alarm.	T#0d0h0m0s0ms	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
QUALITY_DELAYOUT	TIME	The amount of time the input quality must be good to cause the Quality Not Good alarm to clear.	T#0d0h0m0s0ms	False	
HH_LIMIT	REAL	The value (in engineering units) to which the (PV) input is compared for the High High alarm.	0	True	
HH_DEADBAND	REAL	The amount of engineering units by which (PV) must be within the limit for the High High alarm to clear.	0.5	True	
HH_DELAYIN	TIME	The amount of time the input must reach or exceed the High High limit to trip an alarm.	T#0d0h0m0s0ms	False	
HH_DELAYOUT	TIME	The amount of time the input must be within the High High limit and deadband to cause the alarm to clear.	T#0d0h0m0s0ms	False	
LL_LIMIT	REAL	The value (in engineering units) to which the (PV) input is compared for the Low Low alarm.	0	True	
LL_DEADBAND	REAL	The amount of engineering units by which (PV) must be within the limit for the Low Low alarm to clear.	0.5	True	
LL_DELAYIN	TIME	The amount of time the input must reach or exceed the Low Low limit to trip an alarm.	T#0d0h0m0s0ms	False	
LL_DELAYOUT	TIME	The amount of time the input must be within the Low Low limit and deadband to cause the alarm to clear.	T#0d0h0m0s0ms	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
I_ROC_LIMIT	REAL	The value (in engineering) units per minute) to which the (PV) input is compared for the Increasing Rate of Change alarm.	0	True	
I_ROC_DEADBAND	REAL	The amount of engineering units by which (PV) must be within the limit for the Increasing Rate of Change alarm to clear.	0.5	True	
I_ROC_DELAYIN	TIME	The amount of time the input must reach or exceed the Increasing Rate of Change limit to trip an alarm.	T#0d0h0m0s0ms	False	
I_ROC_DELAYOUT	TIME	The amount of time the input must be within the Increasing Rate of Change limit and deadband to cause the alarm to clear.	T#0d0h0m0s0ms	False	
D_ROC_LIMIT	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Decreasing Rate of Change alarm.	0	True	
D_ROC_DEADBAND	REAL	The amount of engineering units by which (PV) must be within the limit for the Decreasing Rate of Change alarm to clear.	0.5	True	
D_ROC_DELAYIN	TIME	The amount of time the input must reach or exceed the Decreasing Rate of Change limit to trip an alarm.	T#0d0h0m0s0ms	False	
D_ROC_DELAYOUT	TIME	The amount of time the input must be within the Decreasing Rate of Change limit and deadband to cause the alarm to clear.	T#0d0h0m0s0ms	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
A_ROC_LIMIT	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Absolute Rate of Change alarm.	0	True	
A_ROC_DEADBAND	REAL	The amount of engineering units by which (PV) must be within the limit for the Absolute Rate of Change alarm to clear.	0.5	True	
A_ROC_DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Rate of change limit to trip an alarm.	T#0d0h0m0s0ms	False	
A_ROC_DELAYOUT	TIME	The amount of time the input must be within the Absolute Rate of Change limit and deadband to cause the alarm to clear.	T#0d0h0m0s0ms	False	
EINL	REAL	The PV low value used for the Out of Range alarm.	0	True	
EINH	REAL	The PV high value used for the Out of Range alarm.	100.0	True	
DISABLE_ON_OOS	BOOL	When set to TRUE, the block is disabled if (OOS) is also TRUE.	False	False	

Note

This list differs from the softlist parameters of the AALM_AFB block in APACS+ as follows:

- The ALARM_TYPE parameters have been eliminated because there are now twelve sets of parameters instead of just four. The type is now part of each parameter name.
- There is now a LIMIT, DEADBAND, DELAYIN, and DELAYOUT parameter for each alarm type that requires it.
- The PCS 7 OS provides a ringback mechanism. The RINGBACK parameters have been removed.

2.3 AM_AFB

2.3.1 AM_AFB: Auto/Manual

Function Block Name	AM_AFB
Function Block Number	FB500

Function

The Auto/Manual Function Block (AM_AFB) regulates the source of a control loop output. Generally, the block outputs either the Input (IN) value in automatic mode or its Internal Output (IOUT) value in manual mode. When the block is in standby mode, the external variable (EXTV) is written to the output (OUT).

The block's IN input is usually connected to the output of a PID block.The OUT value is limited by both low and high limit (LOLIM and HILIM) In/Out parameters. The purpose of these output limits is to prevent the output from being inadvertently driven beyond a process or hardware operating limit. The Internal Output value can be changed via the Computer Output (COUT) or the Auto/Manual function block itself (in track mode). The Computer Output is written to by an operator command. The function block assigns the (OUT) value to (COUT) as well. All values are in Engineering Units.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32). Additionally, it is called in OB100 and in OB102 (see Start-up Characteristics).

Start-up Characteristics

On a warm restart, OB100 is called and parameters retain the values they had immediately before the restart. On a cold restart, OB102 is called and parameters are initialized with the last downloaded values.

The warm and cold startup behavior of OUT is configurable. As delivered, OUT will be forced to 0.0.

To force the OUT parameter to a specified value on a restart, set the RSTRT_OUT_FLAG to true ("Force OUT") and configure RSTRT_OUT_VALUE with the desired forced value. The RSTRT_OUT_VALUE will be used to initialize OUT, if and only if the RSTRT_OUT_FLAG is true and the block restarts in Manual mode (AUTO = "Manual"). If RSTRT_OUT_FLAG is set to false ("Default"), the OUT parameter will be set in accordance with default PCS 7 behavior.

Warning

Using "Default" for the RSTRT_OUT_FLAG could result in your output device retaining its last operating state after a restart. Use "Forced" for the restart flag and configure the associated restart value to ensure a known restart state.

If restart parameter values are changed offline, the configuration must be compiled (changes only) and downloaded (changes only) to the controller in order for the settings to take effect upon restart. If restart parameter values are changed online (via CFC Test mode), the configuration must be compiled (entire program) and downloaded (entire program) to the controller in order for the settings to take effect upon cold restart. This step is not necessary for the online changes to take effect upon a warm restart.

Operator Control and Monitoring (M&C Relevant)

There is no associated faceplate for this block.

2.3.1.1 AM_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
IN	REAL	Input variable. Generally, this input should be the PID function block's output in a simple single loop control application.	0	False	
AUTO	BOOL	Auto Mode input	False	False	
EXTV	REAL	External Variable input	0	False	
STNDBY	BOOL	Standby Mode input	False	False	
RSTRT_OUT_FLAG	BOOL	Default:Retain on WS, Use load val on CS. Force: Use RSTRT_OUT_VALUE on restart	True	False	
RSTRT_OUT_VALUE	REAL	Init restart value for OUT if RSTRT_OUTFLAG true and manual mode	0	False	

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
OUT	REAL	Calculated output variable that has been limited	0	False	
IOUT	REAL	Internal Output	0	False	
LL	BOOL	True when the algorithm is imposing a low limit on (OUT)	False	False	
HL	BOOL	True when the algorithm is imposing a high limit on (OUT)	False	False	

Note

The list differs from the outputs of the AM block in APACS+ as follows:

- COUT has been moved from Output to Input/Output.

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
COUT	REAL	Computer Output, typically written to by an operator command.	0	True	
SLIST_OUT	REAL	Internal copy of IOUT	0	False	
LOLIM	REAL	This is the (OUT) low limit input.	-3.3	True	
HILIM	REAL	This is the (OUT) high limit input.	103.3	True	
INTRK	BOOL	Setting INTRK=TRUE causes the Internal Output to track the (IN) when EXTTRK=FALSE or when (STNDBY)=FALSE and (AUTO)=TRUE.	True	False	
EXTTRK	BOOL	Setting EXTTRK=TRUE causes the Internal Output to track the External Variable when (STNDBY)=TRUE.	True	False	

Note

This list differs from the softlist parameters of the AM block in APACS+ as follows:

- COUT has been moved from Output to Input/Output.

2.4 ANALOG_ALM_AFB

2.4.1 ANALOG_ALM_AFB: Analog Alarm

Function Block Name (compiled version of the Chart)	ANALOG_ALM_AFB
Function Block Number	FB600
Chart Name	ANALOG_ALM
See also, Template Chart (usage example)	**ANALOG_ALM_AT**

Function

The Analog Alarm Function Block (ANALOG_ALM_AFB) is a factory-compiled function block ready-to-use analog indicator with alarms. It is based on the APACS+ ANALOG_ALARM block in the Basic Application Library. As delivered, the PV (Process Variable) input value is monitored for both high and low alarms. The ALARM output value provides the overall alarm status. In addition, there is an alarm output for each analog alarm type. Alarms are annunciated on the HMI.

The ENG_UNITS and DESC In/Outs are for holding values for the engineering units and descriptor, respectively.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32).

Start-up Characteristics

This function block has no special startup properties.

Operator Control and Monitoring (M&C Relevant)

See Faceplate Information

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

ALARM_8P ID	Message #	Message Class	Default Message Text
EV_ID	1	Warning - above	HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID	2	Warning - below	LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID	3	Alarm - above	HIGH_DEV - $$BlockComment$$ - Limit = @6%f@
EV_ID	4	Alarm - below	LOW_DEV - $$BlockComment$$ - Limit = @7%f@
EV_ID	5	Alarm - above	ABSOLUTE_DEV - $$BlockComment$$ - Limit = @8%f@
EV_ID	6	PLC Process Control Message - Failure	OUT_OF_RANGE - $$BlockComment$$
EV_ID	7	PLC Process Control Message - Failure	QUALITY_NOT_GOOD - $$BlockComment$$
EV_ID2	1	Alarm - above	HIGH_ HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID2	2	Alarm - below	LOW_ LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID2	3	Warning - above	INC_ROC - $$BlockComment$$ - Limit = @6%f@
EV_ID2	4	Warning - below	DEC_ROC - $$BlockComment$$ - Limit = @7%f@
EV_ID2	5	Warning - above	ABS_ROC - $$BlockComment$$ - Limit = @8%f@

Assignment of auxiliary process values to the block parameters

ALARM_8P ID	Value	Block Parameter
EV_ID	@1%s@	BA_NA
EV_ID	@2%d@	BA_ID
EV_ID	@3%d@	STEP_NO
EV_ID	@4%f@	H_LIM
EV_ID	@5%f@	L_LIM
EV_ID	@6%f@	HD_LIM
EV_ID	@7%f@	LD_LIM
EV_ID	@8%f@	AD_LIM
EV_ID2	@1%s@	BA_NA
EV_ID2	@2%d@	BA_ID
EV_ID2	@3%d@	STEP_NO
EV_ID2	@4%f@	HH_LIM
EV_ID2	@5%f@	LL_LIM
EV_ID2	@6%f@	IR_LIM
EV_ID2	@7%f@	DR_LIM
EV_ID2	@8%f@	AR_LIM

2.4.1.1 ANALOG_ALM_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
PV	REAL	Process Value	0	True	
DEV	REAL	Deviation	0	False	
H_EN	TRUE	Enable High Alarm	True	True	
L_EN	BOOL	Enable Low Alarm	True	True	
HD_EN	BOOL	Enable High Deviation Alarm	False	True	
LD_EN	BOOL	Enable Low Deviation Alarm	False	True	
AD_EN	BOOL	Enable Absolute Deviation Alarm	False	True	
OOR_EN	BOOL	Enable Out of Range Alarm	False	True	
QUAL_EN	BOOL	Enable Quality Not Good Alarm	False	True	
HH_EN	BOOL	Enable High High Alarm	False	True	
LL_EN	BOOL	Enable Low Low Alarm	False	True	
IR_EN	BOOL	Enable Increasing Rate of Change Alarm	False	True	
DR_EN	BOOL	Enable Decreasing Rate of Change Alarm	False	True	
AR_EN	BOOL	Enable Absolute Rate of Change Alarm	False	True	
OOS	BOOL	Out of Service indicator	False	True	
QBAD	BOOL	Bad quality indicator	False	False	
EV_ID	DWORD	Event ID	Compiler generated for the type, N/A for the chart	False	
EV_ID2	DWORD	Event ID	Compiler generated for the type, N/A for the chart	False	
BA_EN	BOOL	Batch enabled	False	True	
OCCUPIED	BOOL	Batch occupied	False	True	
BA_ID	DWORD	Batch ID number	0	True	
BA_NA	STRING[32]	Batch name	‘‘	True	
STEP_NO	DWORD	Batch step number	0	True	
DecimalPlacesPV	INT	Decimal places displayed in HMI for PV	2	True	
ConfirmChange	BOOL	If TRUE changes from HMI must be confirmed	False	True	

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
ALARM	BOOL	Alarm indicator	False	False	
H_ALM	BOOL	When TRUE, the High alarm is detected and enabled.	False	False	
L_ALM	BOOL	When TRUE, the Low alarm is detected and enabled.	False	False	
HD_ALM	BOOL	When TRUE, the High Deviation alarm is detected and enabled.	False	False	
LD_ALM	BOOL	When TRUE, the Low Deviation alarm is detected and enabled.	False	False	
AD_ALM	BOOL	When TRUE, the Absolute Deviation alarm is detected and enabled.	False	False	
OOR_ALM	BOOL	When TRUE, the Out of Range alarm is detected and enabled.	False	False	
QUAL_ALM	BOOL	When TRUE, the Quality Not Good alarm is detected and enabled.	False	False	
HH_ALM	BOOL	When TRUE, the High High alarm is detected and enabled.	False	False	
LL_ALM	BOOL	When TRUE, the Low Low alarm is detected and enabled.	False	False	
IR_ALM	BOOL	When TRUE, the Increasing Rate of Change alarm is detected and enabled.	False	False	
DR_ALM	BOOL	When TRUE, the Decreasing Rate of Change alarm is detected and enabled.	False	False	
AR_ALM	BOOL	When TRUE, the Absolute Rate of Change alarm is detected and enabled.	False	False	

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
H_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the High alarm.	100	True	
H_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the High alarm to clear.	0.5	True	
H_DELAYIN	TIME	The amount of time the input must reach or exceed the High limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
H_DELAYOUT	TIME	The amount of time the input must be within the High limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
L_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the Low alarm.	0	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
L_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Low alarm to clear.	0.5	True	
L_DELAYIN	TIME	The amount of time the input must reach or exceed the Low limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
L_DELAYOUT	TIME	The amount of time the input must be within the Low limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
HD_LIM	REAL	The value (in engineering units) to which (PV) – (DEV) is compared for the High Deviation alarm.	0	True	
HD_DB	REAL	The amount of engineering units by which (PV) – (DEV) must be within the limit for the High Deviation alarm to clear.	0.5	True	
HD_DELAYIN	TIME	The amount of time the input must reach or exceed the High Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
HD_DELAYOUT	TIME	The amount of time the input must be within the High Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
LD_LIM	REAL	The value (in engineering units) to which (DEV) - (PV) is compared for the Low Deviation alarm.	0	True	
LD_DB	REAL	The amount of engineering units by which (DEV) - (PV) must be within the limit for the Low Deviation alarm to clear.	0.5	True	
LD_DELAYIN	TIME	The amount of time the input must reach or exceed the Low Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
LD_DELAYOUT	TIME	The amount of time the input must be within the Low Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
AD_LIM	REAL	The value (in engineering units) to which ABS((PV) – (DEV)) is compared for the Absolute Deviation alarm.	100	True	
AD_DB	REAL	The amount of engineering units by which ABS((PV) – (DEV)) must be within the limit for the Absolute Deviation alarm to clear.	0.5	True	
AD_DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms

Name	Data Type	Explanation	Initial	OC&M	Valid Values
AD_DELAYOUT	TIME	The amount of time the input must be within the Absolute Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
HH_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the High High alarm.	0	True	
HH_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the High High alarm to clear.	0.5	True	
HH_DELAYIN	TIME	The amount of time the input must reach or exceed the High High limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
HH_DELAYOUT	TIME	The amount of time the input must be within the High High limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
LL_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the Low Low alarm.	0	True	
LL_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Low Low alarm to clear.	0.5	True	
LL_DELAYIN	TIME	The amount of time the input must reach or exceed the Low Low limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
LL_DELAYOUT	TIME	The amount of time the input must be within the Low Low limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
IR_LIM	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Increasing Rate of Change alarm.	0	True	
IR_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Increasing Rate of Change alarm to clear.	0.5	True	
IR_DELAYIN	TIME	The amount of time the input must reach or exceed the Increasing Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
IR_DELAYOUT	TIME	The amount of time the input must be within the Increasing Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms

Name	Data Type	Explanation	Initial	OC&M	Valid Values
DR_LIM	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Decreasing Rate of Change alarm.	0	True	
DR_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Decreasing Rate of Change alarm to clear.	0.5	True	
DR_DELAYIN	TIME	The amount of time the input must reach or exceed the Decreasing Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
DR_DELAYOUT	TIME	The amount of time the input must be within the Decreasing Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
AR_LIM	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Absolute Rate of Change alarm.	0	True	
AR_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Absolute Rate of Change alarm to clear.	0.5	True	
AR_DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
AR_DELAYOUT	TIME	The amount of time the input must be within the Absolute Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
OOR_DELAYIN	TIME	The amount of time the input must be outside the limits EINH, EINL to trip the Out of Range alarm.	T#0MS	False	0 to 24d20h31m23s647ms
OOR_DELAYOUT	TIME	The amount of time the input must be within the limits EINH, EINL to cause the Out of Range alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
QUAL_DELAYIN	TIME	The amount of time the input quality must be bad to trip the Quality Not Good alarm.	T#0MS	False	0 to 24d20h31m23s647ms
QUAL_DELAYOUT	TIME	The amount of time the input quality must be good to cause the Quality Not Good alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
DISABLE_ON_OOS	BOOL	When set to TRUE, the block is disabled if (OOS) is also TRUE.	False	False	
PV_EUMIN	REAL	Engineering Input Low	0	True	
PV_EUMAX	REAL	Engineering Input High	100	True	

2.5 BLK_VLV_1OUT_AFB

2.5.1 BLK_VLV_1OUT: Block Valve with One Output

Function Block Name (compiled version of the Chart)	BLK_VLV_1OUT_AFB
Function Block Number	FB601
Chart Name	BLK_VLV_1OUT
See also, Template Chart (usage example)	**BLK_VLV_1OUT_AT**

Function

The Block Valve with One Output (BLK_VLV_1OUT) is intended for use as an open/close block valve with one output. It is designed for a normally closed valve with auto/manual operation, pulse commands, and interlocking. An alarm is provided for failure to reach the desired state.

The [A_OPEN] and [A_CLSE] inputs are intended to input command pulses from a sequence. The parameter AUTO determines the source of the open/close commands. In auto mode (AUTO=TRUE) the auto inputs [A_OPEN] and [A_CLSE] are active, and in manual mode (AUTO=FALSE), the parameters OPEN and CLOSE are active. OPEN, CLOSE, and AUTO are to be manipulated via the operator interface. OPEN and CLOSE are to be set TRUE; they are reset automatically.

A positive transition of auto open [A_OPEN] or manual open OPEN (depending on the mode), latches the output [OUT]. Interlock [INTRLK] must be TRUE to allow the latch. A positive transition of auto close [A_CLSE] or manual close CLOSE (also mode-dependent) unlatches the output. The internal open and close timers determine if the valve has failed to reach its desired state, and the failure output [FAIL] will become TRUE accordingly, which triggers the alarm.

Although the BLK_VLV_1OUT block was implemented in ladder logic in APACS, it has been implemented in SCL in PCS 7 due to differences between APACS+ and PCS 7 ladder logic. The BLK_VLV_1OUT is a logical and functional equivalent to the corresponding APACS+ ladder networks. Comments in the provided SCL source code relate elements in the PCS 7 SCL code to corresponding elements in the original APACS+ ladder network.

Calling Obs and Start-up

Calling Obs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32). Additionally, it is called in OB100 and in OB102 (see Start-up Characteristics).

Start-up Characteristics

On a warm restart, OB100 is called and parameters retain the values they had immediately before the restart. On a cold restart, OB102 is called and parameters are initialized with the last downloaded values.

The warm and cold startup behavior of AUTO and OUT is configurable, allowing the user to override the default PCS 7 behavior described above if necessary. As delivered, AUTO will be forced to "Manual" and OUT will be forced to "False" on a CPU restart.

To force the AUTO parameter to a specified value on a restart, set the RSTRT_AUTO_FLAG to true ("Force AUTO") and configure RSTRT_AUTO_VALUE with the desired forced value ("Auto" or "Manual"). The RSTRT_AUTO_VALUE will be used to initialize AUTO, if and only if the RSTRT_AUTO_FLAG is true. If RSTRT_AUTO_FLAG is set to false ("Default"), the AUTO parameter will be set in accordance with default PCS 7 behavior.

To force the OUT parameter to a specified value on a restart, set the RSTRT_OUT_FLAG to true ("Force OUT") and configure RSTRT_OUT_VALUE with the desired forced value ("True" or "False"). The RSTRT_OUT_VALUE will be used to initialize OUT, if and only if the RSTRT_OUT_FLAG is true. If RSTRT_OUT_FLAG is set to false ("Default"), the OUT parameter will be set in accordance with default PCS 7 behavior.

Warning

Using "Default" for the RSTRT_OUT_FLAG could result in your output device retaining its last operating state after a restart. Similarly, using "Default" for the RSTRT_AUTO_FLAG could result in your process running in automatic mode after a restart. Use "Forced" for the restart flags and configure the associated restart values to ensure a known restart state.

If restart parameter values are changed offline, the configuration must be compiled (changes only) and downloaded (changes only) to the controller in order for the settings to take effect upon restart. If restart parameter values are changed online (via CFC Test mode), the configuration must be compiled (entire program) and downloaded (entire program) to the controller in order for the settings to take effect upon cold restart. This step is not necessary for the online changes to take effect upon a warm restart.

Operator Control and Monitoring (M&C Relevant)

See Faceplate Information

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

Message #	Message Class	Default Message Text
1	PLC Process Control Message - Failure	Valve Feedback Failure - $$BlockComment$$

Assignment of auxiliary process values to the block parameters

Value	Block Parameter
@1%s@	BA_NA
@2%d@	BA_ID
@3%d@	STEP_NO

2.5.1.1 BLK_VLV_1OUT_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
OP_VER	BOOL	OP_VER feedback to verify that the valve is open	False	True	
CL_VER	BOOL	CL_VER feedback to verify that the valve is closed	False	True	
A_OPEN	BOOL	A false-to-true transition of A_OPEN opens the valve when in AUTO mode (AUTO=true).	False	False	
A_CLSE	BOOL	A false-to-true transition of A_CLSE closes the valve when in AUTO mode (AUTO=true).	False	False	
INTRLK	BOOL	INTRLK forbids activation of the valve	False	True	
SAMPLE_T	REAL	Sample Time (in seconds)	Compiler generated	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
RSTRT_AUTO_FLAG	BOOL	Default retain on WS, Use load val on CS. Force: Use RSTRT_AUTO_VALUE on restart	False	True	
RSTRT_AUTO_VALUE	BOOL	Init restart value for Auto Mode if RSTRT_AUTO_FLAG true	False	False	
RSTRT_OUT_FLAG	BOOL	Default retain on WS, Use load val on CS. Force: Use RSTRT_AUTO_VALUE on restart	False	True	
RSTRT_OUT_VALUE	BOOL	Init restart value for OUT if RSTRT_OUT_FLAG true	False	False	
EV_ID	DWORD	Event ID	Compiler generated	False	
LIOP_MAN_SEL	BOOL	Select AUTO/MANUAL: 1 = Linking, 0 = Operator active	False	False	
AUT_L	BOOL	Linkable input for AUTO/MANUAL mode, 1 = AUTO	False	False	
BA_EN	BOOL	Batch enabled	False	True	
OCCUPIED	BOOL	Batch occupied	False	True	
BA_ID	DWORD	Batch ID number	0	True	
BA_NA	STRING[32]	Batch name	‘‘	True	
STEP_NO	DWORD	Batch step number	0	True	
ConfirmChange	BOOL	If TRUE changes from HMI must be confirmed	False	True	

Note

This list differs from the inputs of the BLK_VLV_1OUT_AFB block in APACS+ as follows:

- Restart parameters (RSTRT_...) did not exist in APACS+.
- EV_ID and SAMPLE_T did not exist in APACS+.

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
OUT	BOOL	OUT is used to actuate the valve	False	True	
ALARM	BOOL	ALARM is used to signa 1 internal logic when the Valve is in alarm	False	False	
FAIL	BOOL	FAIL signals if the valve failed to open or close after an OPEN or CLOSE request	False	True	
TRANS	BOOL	TRANS signals that the valve is in transition (i.e. OP_VER, CL_VER both false)	False	True	

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
AUTO	BOOL	AUTO select auto mode (uses A_OPEN, A_CLSE) vs manual mode (uses OPEN, CLOSE)	False	True	
A1EN	BOOL	A1EN is used to enable/disable the internal alarm	True	True	
A1LIM	BOOL	Alarm 1 limit	True	False	
A1DELAYIN	TIME	Into alarm 1 delay time	T#0MS	False	0 to 24d20h31m 23s647ms
A1DELAYOUT	TIME	Out of alarm 1 delay time	T#0MS	False	0 to 24d20h31m 23s647ms
OOS	BOOL	OOS indicates that the internal alarm is Out of Service	False	True	
DISABLE_ON_OOS	BOOL	Disable on Out of Service	False	False	
OPEN	BOOL	OPEN opens the valve when in Manual mode (AUTO=false)	False	True	
CLOSE	BOOL	CLOSE closes the valve when in Manual mode (AUTO=false)	False	True	
TRAV_T	TIME	TRAV_T sets time limit for confirmation of open/close request	T#10S	True	

2.6 BLK_VLV_2OUT_AFB

2.6.1 BLK_VLV_2OUT: Block Valve with Two Outputs

Function Block Name (compiled version of the Chart)	BLK_VLV_2OUT_AFB
Function Block Number	FB602
Chart Name	BLK_VLV_2OUT
See also, Template Chart (usage example)	**BLK_VLV_2OUT_AT**

Function

The Block Valve with Two Outputs (BLK_VLV_2OUT) is intended for use as an open/close block valve with both an open output and a close output. It is designed for a valve with auto/manual operation, pulse commands, and interlocking. An alarm is provided for failure to reach the desired state.

The [A_OPEN] and [A_CLSE] inputs are intended to input command pulses from a sequence. The parameter AUTO determines the source of the open/close commands. In auto mode (AUTO=TRUE), the auto inputs [A_OPEN] and [A_CLSE] are active, and in manual mode (AUTO=FALSE), the parameters OPEN and CLOSE are active. OPEN, CLOSE, and AUTO are to be manipulated via the operator interface. OPEN and CLOSE are to be set TRUE; they are reset automatically.

A positive transition of auto open [A_OPEN] or manual open OPEN (depending on the mode) latches the internal OPEN_LATCH and pulses the open output [OP_OUT]. Interlock [INTRLK] must be TRUE to allow the latch. A positive transition of auto close [A_CLSE] or manual close CLOSE (also mode-dependent) unlatches the internal OPEN_LATCH and pulses the close output [CL_OUT]. The internal open and close timers pulse the outputs and determine if the valve has failed to reach its desired state, and the failure output [FAIL] will become TRUE accordingly, which triggers the alarm.

Although the BLK_VLV_2OUT block was implemented in ladder logic in APACS, it has been implemented in SCL in PCS 7 due to differences between APACS+ and PCS 7 ladder logic. The BLK_VLV_2OUT is a logical and functional equivalent to the corresponding APACS+ ladder networks. Comments in the provided SCL source code relate elements in the PCS 7 SCL code to corresponding elements in the original APACS+ ladder network.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32). Additionally, it is called in OB100 and in OB102 (see Start-up Characteristics).

Start-up Characteristics

On a warm restart, OB100 is called and parameters retain the values they had immediately before the restart. On a cold restart, OB102 is called and parameters are initialized with the last downloaded values.

The warm and cold startup behavior of AUTO is configurable, allwoing the user to override the default PCS 7 behavior described above if necessary. As delivered, AUTO will be forced to "Manual" and OUT will be forced to "False" on a CPU restart.

To force the AUTO parameter to a specified value on a restart, set the RSTRT_AUTO_FLAG to true ("Force AUTO") and configure RSTRT_AUTO_VALUE with the desired forced value ("Auto" or "Manual"). The RSTRT_AUTO_VALUE will be used to initialize AUTO, if and only if the RSTRT_AUTO_FLAG is true. If RSTRT_AUTO_FLAG is set to false ("Default"), the AUTO parameter will be set in accordance with default PCS 7 behavior.

Warning

Using "Default" for the RSTRT_AUTO_FLAG could result in your process running in automatic mode after a restart. Use "Forced" for the restart flag and configure the associate restart value to ensure a known restart state.

The OP_OUT and CL_OUT outputs will be automatically cleared on a restart.

If restart parameter values are changed offline, the configuration must be compiled (changes only) and downloaded (changes only) to the controller in order for the settings to take effect upon restart. If restart parameter values are changed online (via CFC Test mode), the configuration must be compiled (entire program) and downloaded (entire program) to the controller in order for the settings to take effect upon cold restart. This step is not necessary for the online changes to take effect upon a warm restart.

Operator Control and Monitoring (M&C Relevant)

See Faceplate Information

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

Message #	Message Class	Default Message Text
1	PLC Process Control Message - Failure	Valve Feedback Failure - $$BlockComment$$

Assignment of auxiliary process values to the block parameters

Value	Block Parameter
@1%s@	BA_NA
@2%d@	BA_ID
@3%d@	STEP_NO

2.6.1.1 BLK_VLV_2OUT_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
OP_VER	BOOL	OP_VER feedback to verify that the valve is open	False	True	
CL_VER	BOOL	CL_VER feedback to verify that the valve is closed	False	True	
A_OPEN	BOOL	A_OPEN is intended to allow internal logic to automatically open the valve	False	False	
A_CLSE	BOOL	A_CLSE is intended to allow internal logic to automatically close the valve	False	False	
INTRLK	BOOL	INTRLK forbids activation of the valve	False	True	
SAMPLE_T	REAL	Sample time (in seconds)	Compiler generated	False	
RSTRT_AUTO_FLAG	BOOL	Default retain on WS, Use load val on CS. Force: Use RSTRT_AUTO_VALUE on restart	False	True	
RSTRT_AUTO_VALUE	BOOL	Init restart value for Auto Mode if RSTRT_AUTO_FLAG true	False	False	
EV_ID	DWORD	Event ID	Compiler generated	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
LIOP_MAN_SEL	BOOL	Select AUTO/MANUAL: 1 = Linking, 0 = Operator active	False	False	
AUT_L	BOOL	Linkable input for AUTO/MANUAL mode, 1 = AUTO	False	False	
BA_EN	BOOL	Batch enabled	False	True	
OCCUPIED	BOOL	Batch occupied	False	True	
BA_ID	DWORD	Batch ID number	0	True	
BA_NA	STRING[32]	Batch name	‘‘	True	
STEP_NO	DWORD	Batch step number	0	True	
ConfirmChange	BOOL	If TRUE changes from HMI must be confirmed	False	True	

Note

This list differs from the inputs of the BLK_VLV_2OUT_AFB block in APACS+ as follows:

- Restart parameters (RSTRT_...) did not exist in APACS+.
- EV_ID and SAMPLE_T did not exist in APACS+.

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
OP_OUT	BOOL	OP_OUT is used to request that the valve open	T#0MS	False	0 to 24d20h31m23s647ms
CL_OUT	BOOL	CL_OUT is used to request that the valve close	False	False	
ALARM	BOOL	ALARM is used to signal internal logic when the Valve is in alarm	False	False	
FAIL	BOOL	FAIL signals if the valve failed to open or close after an OPEN or CLOSE request	False	True	
TRANS	BOOL	TRANS signals that the valve is in transition (i.e. OP_VER, CL_VER both false)	False	True	
OUT	BOOL	OUT is used to let the OS 'see' the internal OPEN_LATCH coil	False	True	

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
AUTO	BOOL	AUTO select auto mode (uses A_OPEN, A_CLSE) vs manual mode (uses OPEN, CLOSE)	False	True	
A1EN	BOOL	A1EN is used to enable /disable the internal alarm	True	True	
A1LIM	BOOL	Alarm 1 limit	True	False	
A1DELAYIN	TIME	Into alarm 1 delay time	T#0MS	False	0 to 24d20h31m 23s647ms
A1DELAYOUT	TIME	Out of alarm 1 delay time	T#0MS	False	0 to 24d20h31m 23s647ms
OOS	BOOL	OOS indicates that the internal alarm is Out of Service	False	True	
DISABLE_ON_OOS	BOOL	Indicates that the internal alarm is Out of Service	False	False	
OPEN	BOOL	OPEN opens the valve w hen in Manual mode (AUTO=false)	False	True	
CLOSE	BOOL	Closes the valve when in Manual mode (AUTO=false)	False	True	
TRAV_T	TIME	TRAV_T sets time limit for confirmation of open/close request	T#10S	True	

2.7 CASCADE_AFB

2.7.1 CASCADE: Cascade Controller

Function Block Name (compiled version of the Chart)	CASCADE_AFB
Function Block Number	FB603
Chart Name	CASCADE
See also, Template Chart (usage example)	**CASCADE_AT**

Function

CASCADE_AFB is a compiled version of the CASCADE source chart. Within the chart, there are two connected control loops, primary and secondary. Each contains SETPNT_AFB and PID_AFB blocks along with alarming logic. The secondary loop also contains an AM_AFB block.

The primary controller tracks the secondary process variable [S_PV] when the cascaded loops are in manual mode (AUTO=FALSE) or secondary mode (CASC=FALSE). The PRI_SETPOINT block supplies the setpoint [P_SP_OUT] to the PRI_CONTROLLER block. The PRI_SETPOINT block has a ramping feature for auto mode. The setpoint tracks the primary process variable [P_PV] in manual or secondary mode. The PRI_CONTROLLER block is a PID block; it operates when both AUTO and CASC are TRUE and tracks the secondary process variable [S_PV] if either is FALSE. The PRI_CONTROLLER block output becomes the external setpoint to the secondary control loop.

The SEC_SETPOINT block supplies the setpoint [S_SP] to the SEC_CONTROLLER block. The SEC_SETPOINT block has a ramping feature for secondary mode (CASC=FALSE). In cascade mode (CASC=TRUE), the external setpoint (the output from the primary loop) becomes the secondary controller setpoint, and the SEC_SETPOINT block internal setpoint ISP tracks the external setpoint, thereby ensuring no change in setpoint during transition to secondary mode. The setpoint tracks the secondary process variable [S_PV] in manual mode (AUTO=FALSE).

The SEC_CONTROLLER block is a PID_AFB block; it operates in auto mode (AUTO=TRUE) and tracks the downstream output [OUT] in manual mode (AUTO=FALSE). The SEC_AUTO_MANUAL block provides manual control of the output. In auto mode the SEC_CONTROLLER block output becomes the loop output [OUT]. In manual mode the SEC_AUTO_MANUAL block internal output IOUT becomes the loop output.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32). Additionally, it is called in OB100 and in OB102 (see Start-up Characteristics).

Start-up Characteristics

On a warm restart, OB100 is called and parameters retain the values they had immediately before the restart. On a cold restart, OB102 is called and parameters are initialized with the last downloaded values.

The warm and cold startup behavior of AUTO, P_SP, S_SP, CASC, and OUT is configurable, allowing the user to override the default PCS 7 behavior described above, if necessary. As delivered, AUTO will be forced to "Manual", CASC will be forced to "Cascade" and OUT will be forced to 0.0 on a CPU restart. P_SPand S_SP will be set in accordance with default PCS 7 behavior.

To force the AUTO parameter to a specified value on a restart, set the RSTRT_AUTO_FLAG to true ("Force AUTO") and configure RSTRT_AUTO_VALUE with the desired forced value ("Auto" or "Manual"). The RSTRT_AUTO_VALUE will be used to initialize AUTO, if and only if the RSTRT_AUTO_FLAG is true. If RSTRT_AUTO_FLAG is set to false ("Default"), the AUTO parameter will be set in accordance with default PCS 7 behavior.

To force the P_SP parameter to a specified value on a restart, set the P_RSTRT_SP_FLAG to true ("Force SP") and configure P_RSTRT_SP_VALUE with the desired forced value. The P_RSTRT_SP_VALUE will be used to initialize P_SP, if the P_RSTRT_SP_FLAG is true and either

- the block is forced to start in Auto mode, or
- AUTO is not forced and the block restarts with Auto mode (AUTO="Auto") and in Cascade (CASC = "Cascade").

If P_RSTRT_SP_FLAG is set to false ("Default"), the P_SP parameter will be set in accordance with default
PCS 7 behavior.

To force the S_SP parameter to a specified value on a restart, set the S_RSTRT_SP_FLAG to true ("Force SP") and configure S_RSTRT_SP_VALUE with the desired forced value. The S_RSTRT_SP_VALUE will be used to initialize S_SP, if and only if the S_RSTRT_SP_FLAG is true and the block restarts in Auto mode (AUTO="Auto") and not in Cascade (CASC="Secondary"). If S_RSTRT_SP_FLAG is set to false ("Default"), the S_SP parameter will be set in accordance with default PCS 7 behavior.

To force the CASC parameter to a specified value on a restart, set the RSTRT_CASC_FLAG to true ("Force CASC") and configure RSTRT_CASC_VALUE with the desired forced value ("Secondary or Cascade"). The RSTRT_CASC_VALUE will be used to initialize CASC, if and only if the RSTRT_CASC_FLAG is true. If RSTRT_CASC_FLAG is set to false ("Default"), the CASC parameter will be set in accordance with default PCS 7 behavior. If CASC is true following a restart, then S_SP will be set to the Primary loop's OUT value.

To force the OUT parameter to a specified value on a restart, set the RSTRT_OUT_FLAG to true ("Force OUT") and configure RSTRT_OUT_VALUE with the desired forced value. The RSTRT_OUT_VALUE will be used to initialize OUT, if and only if the RSTRT_OUT_FLAG is true and the block restarts in Manual mode (AUTO = "Manual"). If RSTRT_OUT_FLAG is set to false ("Default"), the OUT parameter will be set in accordance with default PCS 7 behavior.

Warning

Using "Default" for the RSTRT_OUT_FLAG could result in your output device retaining its last operating state after a restart. Similarly, using "Default" for the RSTRT_AUTO_FLAG could result in your process running in automatic mode after a restart. Use "Forced" for the restart flags and configure the associated restart values to ensure a known restart state.

If restart parameter values are changed offline, the configuration must be compiled (changes only) and downloaded (changes only) to the controller in order for the settings to take effect upon restart. If restart parameter values are changed online (via CFC Test mode), the configuration must be compiled (entire program) and downloaded (entire program) to the controller in order for the settings to take effect upon cold restart. This step is not necessary for the online changes to take effect upon a warm restart.

Operator Control and Monitoring (M&C Relevant)

See Faceplate Information

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

ALARM_8P ID	Message #	Message Class	Default Message Text
EV_ID	1	Warning - above	Pri HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID	2	Warning - below	Pri LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID	3	Alarm - above	Pri HIGH_DEV - $$BlockComment$$ - Limit = @6%f@
EV_ID	4	Alarm - below	Pri LOW_DEV - $$BlockComment$$ - Limit = @7%f@
EV_ID	5	Alarm - above	Pri ABSOLUTE_DEV - $$BlockComment$$ - Limit = @8%f@
EV_ID	6	PLC Process Control Message - Failure	Pri OUT_OF_RANGE - $$BlockComment$$
EV_ID	7	PLC Process Control Message - Failure	Pri QUALITY_NOT_GOOD - $$BlockComment$$
EV_ID2	1	Alarm - above	Pri HIGH_ HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID2	2	Alarm - below	Pri LOW_ LOW - $$BlockComment$$ - Limit = @5%f@

ALARM_8P ID	Message #	Message Class	Default Message Text
EV_ID2	3	Warning - above	Pri INC_ROC - $$BlockComment$$ - Limit = @6%f@
EV_ID2	4	Warning - below	Pri DEC_ROC - $$BlockComment$$ - Limit = @7%f@
EV_ID2	5	Warning - above	Pri ABS_ROC - $$BlockComment$$ - Limit = @8%f@
EV_ID3	1	Warning - above	Sec HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID3	2	Warning - below	Sec LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID3	3	Alarm - above	Sec HIGH_DEV - $$BlockComment$$ - Limit = @6%f@
EV_ID3	4	Alarm - below	Sec LOW_DEV - $$BlockComment$$ - Limit = @7%f@
EV_ID3	5	Alarm - above	Sec ABSOLUTE_DEV - $$BlockComment$$ - Limit = @8%f@
EV_ID3	6	PLC Process Control Message - Failure	Sec OUT_OF_RANGE - $$BlockComment$$
EV_ID3	7	PLC Process Control Message - Failure	Sec QUALITY_NOT_GOOD - $$BlockComment$$
EV_ID4	1	Alarm - above	Sec HIGH_ HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID4	2	Alarm - below	Sec LOW_ LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID4	3	Warning - above	Sec INC_ROC - $$BlockComment$$ - Limit = @6%f@
EV_ID4	4	Warning - below	Sec DEC_ROC - $$BlockComment$$ - Limit = @7%f@
EV_ID4	5	Warning - above	Sec ABS_ROC - $$BlockComment$$ - Limit = @8%f@

Assignment of auxiliary process values to the block parameters

ALARM_8P ID	Value	Block Parameter
EV_ID	@1%s@	BA_NA
EV_ID	@2%d@	BA_ID
EV_ID	@3%d@	STEP_NO
EV_ID	@4%f@	P_H_LIM
EV_ID	@5%f@	P_L_LIM
EV_ID	@6%f@	P_HD_LIM
EV_ID	@7%f@	P_LD_LIM
EV_ID	@8%f@	P_AD_LIM
EV_ID2	@1%s@	BA_NA
EV_ID2	@2%d@	BA_ID

ALARM_8P ID	Value	Block Parameter
EV_ID2	@3%d@	STEP_NO
EV_ID2	@4%f@	P_HH_LIM
EV_ID2	@5%f@	P_LL_LIM
EV_ID2	@6%f@	P_IR_LIM
EV_ID2	@7%f@	P_DR_LIM
EV_ID2	@8%f@	P_AR_LIM
EV_ID3	@1%s@	BA_NA
EV_ID3	@2%d@	BA_ID
EV_ID3	@3%d@	STEP_NO
EV_ID3	@4%f@	S_H_LIM
EV_ID3	@5%f@	S_L_LIM
EV_ID3	@6%f@	S_HD_LIM
EV_ID3	@7%f@	S_LD_LIM
EV_ID3	@8%f@	S_AD_LIM
EV_ID4	@1%s@	BA_NA
EV_ID4	@2%d@	BA_ID
EV_ID4	@3%d@	STEP_NO
EV_ID4	@4%f@	S_HH_LIM
EV_ID4	@5%f@	S_LL_LIM
EV_ID4	@6%f@	S_IR_LIM
EV_ID4	@7%f@	S_DR_LIM
EV_ID4	@8%f@	S_AR_LIM

2.7.1.1 CASCADE_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
P_PV	REAL	Primary Process Variable	0	True	
S_PV	REAL	Secondary Process Variable	0	True	
P_INIT	BOOL	Transitions of the INIT input cause the initialization equation to be performed	False	False	
P_TSP	REAL	SP target value for SP ramping	0	True	
P_H_EN	BOOL	Enable primary High Alarm	True	True	
P_L_EN	BOOL	Enable primary Low Alarm	True	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
P_HD_EN	BOOL	Enable primary High Deviation Alarm	False	True	
P_LD_EN	BOOL	Enable primary Low Deviation Alarm	False	True	
P_AD_EN	BOOL	Enable primary Absolute Deviation Alarm	True	True	
P_OOR_EN	BOOL	Enable primary Out of Range Alarm	False	True	
P_QUAL_EN	BOOL	Enable primary Quality Not Good Alarm	False	True	
P_HH_EN	BOOL	Enable primary High High Alarm	False	True	
P_LL_EN	BOOL	Enable primary Low Low Alarm	False	True	
P_IR_EN	BOOL	Enable primary Increasing Rate of Change Alarm	False	True	
P_DR_EN	BOOL	Enable primary Decreasing Rate of Change Alarm	False	True	
P_AR_EN	BOOL	Enable primary Absolute Rate of Change Alarm	False	True	
P_OOS	BOOL	Primary Out of Service indicator	False	True	
P_QBAD	BOOL	Bad quality indicator	False	False	
EV_ID	DWORD	Event ID	Compiler generated for the type or N/A if used as chart *	False	
EV_ID2	DWORD	Event ID	Compiler generated for the type or N/A if used as chart *	False	
EV_ID3	DWORD	Event ID	Compiler generated for the type or N/A if used as chart *	False	
EV_ID4	DWORD	Event ID	Compiler generated for the type or N/A if used as chart *	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
S_TSP	REAL	SP target value for SP ramping	0	True	
S_H_EN	BOOL	Enable secondary High Alarm	True	True	
S_L_EN	BOOL	Enable secondary Low Alarm	True	True	
S_HD_EN	BOOL	Enable secondary High Deviation Alarm	False	True	
S_LD_EN	BOOL	Enable secondary Low Deviation Alarm	False	True	
S_AD_EN	BOOL	Enable secondary Absolute Deviation Alarm	True	True	
S_OOR_EN	BOOL	Enable secondary Out of Range Alarm	False	True	
S_QUAL_EN	BOOL	Enable secondary Quality Not Good Alarm	False	True	
S_HH_EN	BOOL	Enable secondary High High Alarm	False	True	
S_LL_EN	BOOL	Enable secondary Low Low Alarm	False	True	
S_IR_EN	BOOL	Enable secondary Increasing Rate of Change Alarm	False	True	
S_DR_EN	BOOL	Enable secondary Decreasing Rate of Change Alarm	False	True	
S_AR_EN	BOOL	Enable secondary Absolute Rate of Change Alarm	False	True	
S_OOS	BOOL	Secondary Out of Service indicator	False	True	
S_QBAD	BOOL	Bad quality indicator	False	False	
SAMPLE_T	REAL	Scan Rate. Use only when compiling CFC as block type.	Compiler generated for the type or N/A if used as chart *	False	
RSTRT_AUTO_FLAG	BOOL	Default:Retain on WS, Use load val on CS. Force:Use RSTRT_AUTO_VALUE on restart	True	False	
RSTRT_AUTO_VALUE	BOOL	Init restart value for Auto Mode if RSTRT_AUTO_FLAG true	False	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
P_RSTRT_SP_FLAG	BOOL	Default:Retain on WS, Use load val on CS. Force:Use RSTRT_SP_VALUE on restart.	False	False	
P_RSTRT_SP_VALUE	REAL	Init restart value for SP if RSTRT_SP_FLAG true and Auto mode	0	False	
RSTRT_CASC_FLAG	BOOL	Default:Retain on WS, Use load val on CS. Force:Use RSTRT_EXT_VALUE on restart.	True	False	
RSTRT_CASC_VALUE	BOOL	Init restart value for EXTernal if RSTRT_EXT_FLAG true	False	False	
S_RSTRT_SP_FLAG	BOOL	Default:Retain on WS, Use load val on CS. Force:Use RSTRT_SP_VALUE on restart.	False	False	
S_RSTRT_SP_VALUE	REAL	Init restart value for SP if RSTRT_SP_FLAG true and Auto mode	0	False	
RSTRT_OUT_FLAG	BOOL	Default:Retain on WS, Use load val on CS. Force:Use RSTRT_OUT_VALUE on restart	True	False	
RSTRT_OUT_VALUE	REAL	Init restart value for OUT if RSTRT_OUT_FLAG true and manual mode	0	False	
BA_EN	BOOL	Batch enabled	False	True	
OCCUPIED	BOOL	Batch occupied	False	True	
BA_ID	DWORD	Batch ID number	0	True	
BA_NA	STRING[32]	Batch name	‘‘	True	
STEP_NO	DWORD	Batch step number	0	True	
DecimalPlacesP_PV	INT	Decimal places displayed in HMI for Primary PV, SP etc.	2	True	
DecimalPlacesS_PV	INT	Decimal places displayed in HMI for Secondary PV, SPetc.	2	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
DecimalPlacesOUT	INT	Decimal places displayed in HMI for OUT, OUT Limits	2	True	
ConfirmChange	BOOL	If TRUE changes from HMI must be confirmed	False	True	
LIOP_MAN_SEL	BOOL	Select AUTO/MANUAL: 1 = Linking, 0 = Operator active	0	False	
AUT_L	BOOL	Linkable input for AUTO/MANUAL mode, 1 = AUTO	0	False	
LIOP_P_ISP_SEL	BOOL	Select Primary internal SP: 1 = from Link, 0 = from Operator	0	False	
P_SP_L	REAL	Linkable input for Primary internal Setpoint	0.0	False	
LIOP_S_ISP_SEL	BOOL	Select Secondary internal SP: 1 = from Link, 0 = from Operator	0	False	
S_SP_L	REAL	Linkable input for Secondary internal Setpoint	0.0	False	
LIOP_EXT_SEL	BOOL	Select External/Internal SP flag: 1 = from Link, 0 = from Operator	0	False	
EXTRNL_L	BOOL	Linkable input to select External/Internal SP flag, 1=Extrnl	0	False	
LIOP_OUT_SEL	BOOL	Select writing to OUT: 1 = from Link, 0 = from Operator	0	False	
OUT_L	REAL	Linkable input to be written to Secondary OUT	0.0	False	

Note

This list differs significantly from the inputs of the CASCADE_AFB block in APACS+. In the APACS+ version, only the P_PV and S_PV inputs are present (and in APACS+, they are named PRI_PV and SEC_PV, respectively).

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
P_SP_OUT	REAL	Calculated set point value	0	False	
S_SP_OUT	REAL	Calculated set point value	0	False	
OUT	REAL	Calculated output variable that has been limited	0	False	
ALARM	BOOL	Alarm Status	False	False	
S_ESP	REAL	Provides S_ESP tag for the OS	0	True	
P_H_ALM	BOOL	When TRUE, the primary High alarm is detected and enabled.	False	False	
P_L_ALM	BOOL	When TRUE, the primary Low alarm is detected and enabled.	False	False	
P_HD_ALM	BOOL	When TRUE, the primary High Deviation alarm is detected and enabled.	False	False	
P_LD_ALM	BOOL	When TRUE, the primary Low Deviation alarm is detected and enabled.	False	False	
P_AD_ALM	BOOL	When TRUE, the primary Absolute Deviation alarm is detected and enabled.	False	False	
P_OOR_ALM	BOOL	When TRUE, the primary Out of Range alarm is detected and enabled.	False	False	
P_QUAL_ALM	BOOL	When TRUE, the primary Quality Not Good alarm is detected and enabled.	False	False	
P_HH_ALM	BOOL	When TRUE, the primary High High alarm is detected and enabled.	False	False	
P_LL_ALM	BOOL	When TRUE, the primary Low Low alarm is detected and enabled.	False	False	
P_IR_ALM	BOOL	When TRUE, the primary Increasing Rate of Change alarm is detected and enabled.	False	False	
P_DR_ALM	BOOL	When TRUE, the primary Decreasing Rate of Change alarm is detected and enabled.	False	False	
P_AR_ALM	BOOL	When TRUE, the primary Absolute Rate of Change alarm is detected and enabled.	False	False	
S_H_ALM	BOOL	When TRUE, the secondary High alarm is detected and enabled.	False	False	
S_L_ALM	BOOL	When TRUE, the secondary Low alarm is detected and enabled.	False	False	
S_HD_ALM	BOOL	When TRUE, the secondary High Deviation alarm is detected and enabled.	False	False	
S_LD_ALM	BOOL	When TRUE, the secondary Low Deviation alarm is detected and enabled.	False	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
S_AD_ALM	BOOL	When TRUE, the secondary Absolute Deviation alarm is detected and enabled.	False	False	
S_OOR_ALM	BOOL	When TRUE, the secondary Out of Range alarm is detected and enabled.	False	False	
S_QUAL_ALM	BOOL	When TRUE, the secondary Quality Not Good alarm is detected and enabled.	False	False	
S_HH_ALM	BOOL	When TRUE, the secondary High High alarm is detected and enabled.	False	False	
S_LL_ALM	BOOL	When TRUE, the secondary Low Low alarm is detected and enabled.	False	False	
S_IR_ALM	BOOL	When TRUE, the secondary Increasing Rate of Change alarm is detected and enabled.	False	False	
S_DR_ALM	BOOL	When TRUE, the secondary Decreasing Rate of Change alarm is detected and enabled.	False	False	
S_AR_ALM	BOOL	When TRUE, the secondary Absolute Rate of Change alarm is detected and enabled.	False	False	
S_VL_EUMIN_OUT	REAL	Copy of secondary VL_EUMIN Input	0	False	
S_VL_EUMAX_OUT	REAL	Copy of secondary VL_EUMAX Input	0	False	

Note

The P_SP and S_SP outputs in APACS+ have been converted to P_SP_OUT and S_SP_OUT, respectively. They are used to interconnect the set-point value to other entities on a chart. The OS interaction is handled by the P_SP and S_SP InOut parameters.

The hidden S_ESP output was also added to support OS interaction. The S_ESP output is driven by the primary PID's OUT output.

Alarm outputs have been added for each analog alarm type.

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
CASC	BOOL	Cascade mode	False	True	
AUTO	BOOL	Auto/manual mode control	False	True	
P_SP	REAL	Computer set point value	0	True	
P_RAMP	BOOL	SP ramping on/off mode control	False	True	
P_R_TIME	TIME	Ramp time	T#0MS	True	0 to 24d20h31m 23s647ms
P_SPL	REAL	Set point low limit	0	True	
P_SPH	REAL	Set point high limit	100	True	
P_R_RATE	REAL	Ramp rate (in engineering units per min)	0	True	
P_USE_RR	BOOL	Ramp mode control. Use RTIME or RRATE to control ramp	False	True	
P_PV_TRACK	BOOL	Internal setpoint tracking control. When Track, internal setpoint tracks PV	True	True	
P_PG	REAL	Proportional Gain	0.001	True	0.001 to 100.0
P_TI	REAL	Time-Integral Minutes per Reset	4000	True	0.0 to 4000.0
P_TD	REAL	Time-Derivative Minutes	0	True	0.00 to 100.0
P_DG	REAL	Derivative Gain	10	True	2.00 to 20.00
P_DIR	BOOL	Reverse or Direct action	False	True	
P_ADAPTIVE_PG_INIT	BOOL	Adaptive Gain Init flag	True	False	
P_PV_EUMIN	REAL	Engineering unit: Input, Low	0	True	
P_PV_EUMAX	REAL	Engineering unit: Input, High	100	True	
P_VL_EUMIN	REAL	Engineering unit: Output, Low	0	True	
P_VL_EUMAX	REAL	Engineering unit: Output, High	100	True	
P_H_LIM	REAL	The value (in engineering units) to which the (P_PV) input is compared for the High alarm.	100	True	
P_H_DB	REAL	The amount of engineering units by which (P_PV) must be within the limit for the High alarm to clear.	0.5	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
P_H_DELAYIN	TIME	The amount of time the input must reach or exceed the High limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
P_H_DELAYOUT	TIME	The amount of time the input must be within the High limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
P_L_LIM	REAL	The value (in engineering units) to which the (P_PV) input is compared for the Low alarm.	0	True	
P_L_DB	REAL	The amount of engineering units by which (P_PV) must be within the limit for the Low alarm to clear.	0.5	True	
P_L_DELAYIN	TIME	The amount of time the input must reach or exceed the Low limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
P_L_DELAYOUT	TIME	The amount of time the input must be within the Low limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
P_HD_LIM	REAL	The value (in engineering units) to which (P_PV) – (DEV) is compared for the High Deviation alarm.	0	True	
P_HD_DB	REAL	The amount of engineering units by which (P_PV) – (DEV) must be within the limit for the High Deviation alarm to clear.	0.5	True	
P_HD_DELAYIN	TIME	The amount of time the input must reach or exceed the High Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
P_HD_DELAYOUT	TIME	The amount of time the input must be within the High Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
P_LD_LIM	REAL	The value (in engineering units) to which (DEV) - (P_PV) is compared for the Low Deviation alarm.	0	True	
P_LD_DB	REAL	The amount of engineering units by which (DEV) - (P_PV) must be within the limit for the Low Deviation alarm to clear.	0.5	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
P_LD_DELAYIN	TIME	The amount of time the input must reach or exceed the Low Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
P_LD_DELAYOUT	TIME	The amount of time the input must be within the Low Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
P_AD_LIM	REAL	The value (in engineering units) to which ABS((P_PV) – (DEV)) is compared for the Absolute Deviation alarm.	100	True	
P_AD_DB	REAL	The amount of engineering units by which ABS((P_PV) – (DEV)) must be within the limit for the Absolute Deviation alarm to clear.	0.5	True	
P_AD_DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
P_AD_DELAYOUT	TIME	The amount of time the input must be within the Absolute Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
P_HH_LIM	REAL	The value (in engineering units) to which the (P_PV) input is compared for the High High alarm.	0	True	
P_HH_DB	REAL	The amount of engineering units by which (P_PV) must be within the limit for the High High alarm to clear.	0.5	True	
P_HH_DELAYIN	TIME	The amount of time the input must reach or exceed the High High limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
P_HH_DELAYOUT	TIME	The amount of time the input must be within the High High limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
P_LL_LIM	REAL	The value (in engineering units) to which the (P_PV) input is compared for the Low Low alarm.	0	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
P_LL_DB	REAL	The amount of engineering units by which (P_PV) must be within the limit for the Low Low alarm to clear.	0.5	True	
P_LL_DELAYIN	TIME	The amount of time the input must reach or exceed the Low Low limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
P_LL_DELAYOUT	TIME	The amount of time the input must be within the Low Low limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
P_IR_LIM	REAL	The value (in engineering units per minute) to which the (P_PV) input is compared for the Increasing Rate of Change alarm.	0	True	
P_IR_DB	REAL	The amount of engineering units by which (P_PV) must be within the limit for the Increasing Rate of Change alarm to clear.	0.5	True	
P_IR_DELAYIN	TIME	The amount of time the input must reach or exceed the Increasing Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
P_IR_DELAYOUT	TIME	The amount of time the input must be within the Increasing Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
P_DR_LIM	REAL	The value (in engineering units per minute) to which the (P_PV) input is compared for the Decreasing Rate of Change alarm.	0	True	
P_DR_DB	REAL	The amount of engineering units by which (P_PV) must be within the limit for the Decreasing Rate of Change alarm to clear.	0.5	True	
P_DR_DELAYIN	TIME	The amount of time the input must reach or exceed the Decreasing Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms

Name	Data Type	Explanation	Initial	OC&M	Valid Values
P_DR_DELAYOUT	TIME	The amount of time the input must be within the Decreasing Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
P_AR_LIM	REAL	The value (in engineering units per minute) to which the (P_PV) input is compared for the Absolute Rate of Change alarm.	0	True	
P_AR_DB	REAL	The amount of engineering units by which (P_PV) must be within the limit for the Absolute Rate of Change alarm to clear.	0.5	True	
P_AR_DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
P_AR_DELAYOUT	TIME	The amount of time the input must be within the Absolute Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
P_OOR_DELAYIN	TIME	The amount of time the input must be outside the limits EINH, EINL to trip the Out of Range alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
P_OOR_DELAYOUT	TIME	The amount of time the input must be within the limits EINH, EINL to cause the Out of Range alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
P_QUAL_DELAYIN	TIME	The amount of time the input quality must be bad to trip the Quality Not Good alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
P_QUAL_DELAYOUT	TIME	The amount of time the input quality must be good to cause the Quality Not Good alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
P_DISABLE_ON_OOS	BOOL	When set to TRUE, the block is disabled if (OOS) is also TRUE.	False	False	
S_SP	REAL	Computer set point value	0	True	
S_RAMP	BOOL	SP ramping on/off mode control	False	True	
S_R_TIME	TIME	Ramp time	T#0MS	True	0 to 24d20h31m 23s647ms
S_SPL	REAL	Set point low limit	0	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
S_SPH	REAL	Set point high limit	100	True	
S_R_RATE	REAL	Ramp rate (in engineering units per min)	0	True	
S_USE_RR	BOOL	Ramp mode control. Use RTIME or RRATE to control ramp	False	True	
S_PV_TRACK	BOOL	Internal setpoint tracking control. When Track, internal setpoint tracks PV	True	True	
S_PG	REAL	Proportional Gain	0.001	True	0.001 to 100.0
S_TI	REAL	Time-Integral Minutes per Reset	4000	True	0.0 to 4000.0
S_TD	REAL	Time-Derivative Minutes	0	True	0.00 to 100.0
S_DG	REAL	Derivative Gain	10	True	2.00 to 20.00
S_DIR	BOOL	Reverse or Direct action	False	True	
S_ADAPTIVE_PG_INIT	BOOL	Adaptive Gain Init flag	True	False	
S_PV_EUMIN	REAL	Engineering unit: Input, Low	0	True	
S_PV_EUMAX	REAL	Engineering unit: Input, High	100	True	
S_VL_EUMIN	REAL	Engineering unit: Output, Low	0	True	
S_VL_EUMAX	REAL	Engineering unit: Output, High	100	True	
S_VL	REAL	Computer Output, typically written to by an operator command	0	True	
S_VLL	REAL	OUT low limit input	-3.3	True	
S_VLH	REAL	OUT high limit input	103.3	True	
S_H_LIM	REAL	The value (in engineering units) to which the (S_PV) input is compared for the High alarm.	100	True	
S_H_DB	REAL	The amount of engineering units by which (S_PV) must be within the limit for the High alarm to clear.	0.5	True	
S_H_DELAYIN	TIME	The amount of time the input must reach or exceed the High limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
S_H_DELAYOUT	TIME	The amount of time the input must be within the High limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms

Name	Data Type	Explanation	Initial	OC&M	Valid Values
S_L_LIM	REAL	The value (in engineering units) to which the (S_PV) input is compared for the Low alarm.	0	True	
S_L_DB	REAL	The amount of engineering units by which (S_PV) must be within the limit for the Low alarm to clear.	0.5	True	
S_L_DELAYIN	TIME	The amount of time the input must reach or exceed the Low limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
S_L_DELAYOUT	TIME	The amount of time the input must be within the Low limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
S_HD_LIM	REAL	The value (in engineering units) to which (S_PV) – (DEV) is compared for the High Deviation alarm.	0	True	
S_HD_DB	REAL	The amount of engineering units by which (S_PV) – (DEV) must be within the limit for the High Deviation alarm to clear.	0.5	True	
S_HD_DELAYIN	TIME	The amount of time the input must reach or exceed the High Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
S_HD_DELAYOUT	TIME	The amount of time the input must be within the High Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
S_LD_LIM	REAL	The value (in engineering units) to which (DEV) - (S_PV) is compared for the Low Deviation alarm.	0	True	
S_LD_DB	REAL	The amount of engineering units by which (DEV) - (S_PV) must be within the limit for the Low Deviation alarm to clear.	0.5	True	
S_LD_DELAYIN	TIME	The amount of time the input must reach or exceed the Low Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
S_LD_DELAYOUT	TIME	The amount of time the input must be within the Low Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms

Name	Data Type	Explanation	Initial	OC&M	Valid Values
S_AD_LIM	REAL	The value (in engineering units) to which ABS((S_PV) – (DEV)) is compared for the Absolute Deviation alarm.	100	True	
S_AD_DB	REAL	The amount of engineering units by which ABS((S_PV) – (DEV)) must be within the limit for the Absolute Deviation alarm to clear.	0.5	True	
S_AD_DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
S_AD_DELAYOUT	TIME	The amount of time the input must be within the Absolute Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
S_HH_LIM	REAL	The value (in engineering units) to which the (S_PV) input is compared for the High High alarm.	0	True	
S_HH_DB	REAL	The amount of engineering units by which (S_PV) must be within the limit for the High High alarm to clear.	0.5	True	
S_HH_DELAYIN	TIME	The amount of time the input must reach or exceed the High High limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
S_HH_DELAYOUT	TIME	The amount of time the input must be within the High High limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
S_LL_LIM	REAL	The value (in engineering units) to which the (S_PV) input is compared for the Low Low alarm.	0	True	
S_LL_DB	REAL	The amount of engineering units by which (S_PV) must be within the limit for the Low Low alarm to clear.	0.5	True	
S_LL_DELAYIN	TIME	The amount of time the input must reach or exceed the Low Low limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms

Name	Data Type	Explanation	Initial	OC&M	Valid Values
S_LL_DELAYOUT	TIME	The amount of time the input must be within the Low Low limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
S_IR_LIM	REAL	The value (in engineering units per minute) to which the (S_PV) input is compared for the Increasing Rate of Change alarm.	0	True	
S_IR_DB	REAL	The amount of engineering units by which (S_PV) must be within the limit for the Increasing Rate of Change alarm to clear.	0.5	True	
S_IR_DELAYIN	TIME	The amount of time the input must reach or exceed the Increasing Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
S_IR_DELAYOUT	TIME	The amount of time the input must be within the Increasing Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
S_DR_LIM	REAL	The value (in engineering units per minute) to which the (S_PV) input is compared for the Decreasing Rate of Change alarm.	0	True	
S_DR_DB	REAL	The amount of engineering units by which (S_PV) must be within the limit for the Decreasing Rate of Change alarm to clear.	0.5	True	
S_DR_DELAYIN	TIME	The amount of time the input must reach or exceed the Decreasing Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
S_DR_DELAYOUT	TIME	The amount of time the input must be within the Decreasing Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
S_AR_LIM	REAL	The value (in engineering units per minute) to which the (S_PV) input is compared for the Absolute Rate of Change alarm.	0	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
S_AR_DB	REAL	The amount of engineering units by which (S_PV) must be within the limit for the Absolute Rate of Change alarm to clear.	0.5	True	
S_AR_DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
S_AR_DELAYOUT	TIME	The amount of time the input must be within the Absolute Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
S_OOR_DELAYIN	TIME	The amount of time the input must be outside the limits EINH, EINL to trip the Out of Range alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
S_OOR_DELAYOUT	TIME	The amount of time the input must be within the limits EINH, EINL to cause the Out of Range alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
S_QUAL_DELAYIN	TIME	The amount of time the input quality must be bad to trip the Quality Not Good alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
S_QUAL_DELAYOUT	TIME	The amount of time the input quality must be good to cause the Quality Not Good alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
S_DISABLE_ON_OOS	BOOL	When set to TRUE, the block is disabled if (OOS) is also TRUE.	False	False	

2.8 DALM_AFB

2.8.1 DALM_AFB: Discrete Alarm

Function Block Name	DALM_AFB
Function Block Number	FB504

Function

The Discrete Alarm Function Block (DALM_AFB) is intended for detecting and signaling abnormal discrete data values. This block is used to detect the discrete states (TRUE or FALSE) of up to eight inputs. When an alarm state is detected, an ALM output value and the corresponding NAK output value are set to TRUE. Once the alarm has been acknowledged, the NAK output value is set to FALSE. A given ALM output value is FALSE when its alarm state does not match the corresponding IN input value.

The Out-Of-Service (OOS) output value indicates to monitoring devices when the alarm block has been put in the Out-Of-Service state.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32).

Start-up Characteristics

This function block has no special startup properties.

Operator Control and Monitoring (M&C Relevant)

There is no associated faceplate for this block.

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

Message #	Message Class	Default Message Text
1	PLC Process Control Message - Failure	DISCRETE - $$BlockComment$$
2	PLC Process Control Message - Failure	DISCRETE - $$BlockComment$$
3	PLC Process Control Message - Failure	DISCRETE - $$BlockComment$$
4	PLC Process Control Message - Failure	DISCRETE - $$BlockComment$$
5	PLC Process Control Message - Failure	DISCRETE - $$BlockComment$$
6	PLC Process Control Message - Failure	DISCRETE - $$BlockComment$$
7	PLC Process Control Message - Failure	DISCRETE - $$BlockComment$$
8	PLC Process Control Message - Failure	DISCRETE - $$BlockComment$$

Assignment of auxiliary process values to the block parameters

Value	Block Parameter
@1%s@	BA_NA
@2%d@	BA_ID
@3%d@	STEP_NO

2.8.1.1 DALM_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
IN1	BOOL	This is the value to be monitored for alarm1.	False	True	
EN1	BOOL	When TRUE, alarm1 is enabled.	False	True	
IN2	BOOL	This is the value to be monitored for alarm2.	False	False	
EN2	BOOL	When TRUE, alarm2 is enabled.	False	False	
IN3	BOOL	This is the value to be monitored for alarm3.	False	False	
EN3	BOOL	When TRUE, alarm3 is enabled.	False	False	
IN4	BOOL	This is the value to be monitored for alarm4.	False	False	
EN4	BOOL	When TRUE, alarm4 is enabled.	False	False	
IN5	BOOL	This is the value to be monitored for alarm5.	False	False	
EN5	BOOL	When TRUE, alarm5 is enabled.	False	False	
IN6	BOOL	This is the value to be monitored for alarm6.	False	False	
EN6	BOOL	When TRUE, alarm6 is enabled.	False	False	
IN7	BOOL	This is the value to be monitored for alarm7.	False	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
EN7	BOOL	When TRUE, alarm7 is enabled	False	False	
IN8	BOOL	This is the value to be monitored for alarm8.	False	False	
EN8	BOOL	When TRUE, alarm8 is enabled.	False	False	
OOS	BOOL	Out-Of-Sevice indicator.	False	True	
EV_ID	DWORD	Event ID – This value is a unique message number supplied by PCS 7. There is one for each message (i.e. ALARM_8P) block called.	None	False	
BA_NA	STRING[32]	Batch name	‘‘	False	
BA_ID	DWORD	Batch ID number	0	False	
STEP_NO	DWORD	Batch step number	0	False	

Note

This list differs from the inputs of the DALM_AFB block in APACS+ as follows:

- There are now eight inputs and enables.
- Only the first alarm’s input and enable will be visible, so that the block will look like the APACS+ HLLDA1 block.
- The parameter EV_ID was added to support the message system.
- Batch parameters BA_NA, BA_ID, STEP_NO are added to pass batch values in alarm messages.

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
ALM1	BOOL	When TRUE, alarm1 is detected and enabled.	False	False	
NAK1	BOOL	When TRUE, alarm1 alarm is tripped but not acknowledged.	False	False	
ALM2	BOOL	When TRUE, alarm2 is detected and enabled.	False	False	
NAK2	BOOL	When TRUE, alarm2 is tripped but not acknowledged.	False	False	
ALM3	BOOL	When TRUE, alarm3 is detected and enabled.	False	False	
NAK3	BOOL	When TRUE, alarm3 is tripped but not acknowledged.	False	False	
ALM4	BOOL	When TRUE, alarm4 is detected and enabled.	False	False	
NAK4	BOOL	When TRUE, alarm4 is tripped but not acknowledged.	False	False	
ALM5	BOOL	When TRUE, alarm5 is detected and enabled.	False	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
NAK5	BOOL	When TRUE, alarm5 is tripped but not acknowledged.	False	False	
ALM6	BOOL	When TRUE, alarm6 is detected and enabled.	False	False	
NAK6	BOOL	When TRUE, alarm6 is tripped but not acknowledged.	False	False	
ALM7	BOOL	When TRUE, alarm7 is detected and enabled.	False	False	
NAK7	BOOL	When TRUE, alarm7 is tripped but not acknowledged.	False	False	
ALM8	BOOL	When TRUE, alarm8 is detected and enabled.	False	False	
NAK8	BOOL	When TRUE, alarm8 is tripped but not acknowledged.	False	False	
MSG_ERR	BOOL	ALARM_8P error	False	False	
MSG_STAT	WORD	ALARM_8P status	False	W#16#0	

Note

This list differs from the outputs of the DALM_AFB block in APACS+ as follows:

- There are now eight alarms and NAKs.
- Only the first alarm's output and NAK are visible. This makes the block look similar to the APACS+ HLLDA1 block.
- The STATUS and CMD outputs have been removed.
- MSG_ERR and MSG_STAT have been added to provide the status of the message processing. For specific values, see the online help for ALARM_8P.

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
ALARM1_LIMIT	BOOL	The value to which the IN1 value is compared for alarm1.	False	True	
ALARM1_DELAYIN	TIME	The amount of time the input must remain at the limit to trip alarm1.	T#0d0h0m0s0ms	False	
ALARM1_DELAYOUT	TIME	The amount of time the input must remain not at the limit to cause alarm1 to clear.	T#0d0h0m0s0ms	False	
ALARM2_LIMIT	BOOL	The value to which IN2 is compared for alarm2.	False	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
ALARM2_DELAYIN	TIME	The amount of time the input must remain at the limit to trip alarm2.	T#0d0h0m0s0ms	False	
ALARM2_DELAYOUT	TIME	The amount of time the iput must remain not at the limit to cause alarm2 to clear.	T#0d0h0m0s0ms	False	
ALARM3_LIMIT	BOOL	The value to which the IN3 is compared for alarm3.	False	False	
ALARM3_DELAYIN	TIME	The amount of time the input must remain at the limit to trip alarm3.	T#0d0h0m0s0ms	False	
ALARM3_DELAYOUT	TIME	The amount of time the input must remain not at the limit to cause alarm3 to clear.	T#0d0h0m0s0ms	False	
ALARM4_LIMIT	BOOL	The value to which IN4 is compared for alarm4.	False	False	
ALARM4_DELAYIN	TIME	The amount of time the input must remain at the limit to trip alarm4.	T#0d0h0m0s0ms	False	
ALARM4_DELAYOUT	TIME	The amount of time the input must remain not at the limit to cause alarm4 to clear.	T#0d0h0m0s0ms	False	
ALARM5_LIMIT	BOOL	The value to which IN5 is compared for alarm5.	False	False	
ALARM5_DELAYIN	TIME	The amount of time the input must remain at the limit to trip alarm5.	T#0d0h0m0s0ms	False	
ALARM5_DELAYOUT	TIME	The amount of time the input must remain not at the limit to cause alarm5 to clear.	T#0d0h0m0s0ms	False	
ALARM6_LIMIT	BOOL	The value to which IN6 is compared for alarm6.	False	False	
ALARM6_DELAYIN	TIME	The amount of time the input must remain at the limit to trip alarm6.	T#0d0h0m0s0ms	False	
ALARM6_DELAYOUT	TIME	The amount of time the input must remain not at the limit to cause alarm6 to clear.	T#0d0h0m0s0ms	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
ALARM7_LIMIT	BOOL	The value to which IN7 is compared for alarm7.	False	False	
ALARM7_DELAYIN	TIME	The amount of time the input must remain at the limit to trip alarm7.	T#0d0h0m0s0ms	False	
ALARM7_DELAYOUT	TIME	The amount of time the input must remain not at the limit to cause alarm7 to clear.	T#0d0h0m0s0ms	False	
ALARM8_LIMIT	BOOL	The value to which IN8 is compared for alarm8.	False	False	
ALARM8_DELAYIN	TIME	The amount of time the input must remain at the limit to trip alarm8.	T#0d0h0m0s0ms	False	
ALARM8_DELAYOUT	TIME	The amount of time the input must remain not at the limit to cause alarm8 to clear.	T#0d0h0m0s0ms	False	
DISABLE_ON_OOS	BOOL	When set to TRUE, the block is disabled if (OOS) is also TRUE.	False		

Note

This list differs from the softlist parameters of the DALM_AFB block in APACS+ as follows:

- The PCS 7 OS provides a ringback mechanism, so the ringback parameters have been removed.
- There are now eight LIMIT, DELAYIN, and DELAYOUT parameters.

2.9 DISCRETE_ALM_AFB

2.9.1 DISCRETE_ALM_AFB

Function Block Name (compiled version of the Chart)	DISCRETE_ALM_AFB
Function Block Number	FB604
Chart Name	DISCRETE_ALM
See also, Template Chart (usage example)	**DISCRETE_ALM_AT**

Function

The Discrete Alarm (DISCRETE_ALM_AFB) block is a factory-compiled function block. It is intended for use as a discrete indicator with alarm. The ALARM output provides the alarm status of the block. The block signals an alarm on the HMI when the discrete input (DIN) is in an undesired state.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32).

Start-up Characteristics

This function block has no special startup properties.

Operator Control and Monitoring (M&C Relevant)

See Faceplate Information

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

Message #	Message Class	Default Message Text
1	PLC Process Control Message - Failure	DISCRETE - $$BlockComment$$

Assignment of auxiliary process values to the block parameters

Value	Block Parameter
@1%s@	BA_NA
@2%d@	BA_ID
@3%d@	STEP_NO

2.9.1.1 DISCRETE_ALM_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
DIN	BOOL	The discrete parameter being monitored.	False	True	
ENABLE	BOOL	Enable Alarm 1	True	True	
OOS	BOOL	Out of Service Indicator	False	True	
EV_ID	DWORD	Event ID	Compiler generated for the type, N/A for the chart	False	
BA_EN	BOOL	Batch enabled	False	True	
OCCUPIED	BOOL	Batch occupied	False	True	
BA_ID	DWORD	Batch ID number	0	True	
STEP_NO	DWORD	Batch step number	0	True	
BA_NA	STRING[32]	Batch name	‘‘	True	
ConfirmChange	BOOL	If TRUE changes from HMI must be confirmed	False	True	

Note

The APACS+ list only includes the DI input. Other inputs are hidden from view, so that from a user perspective, the DISCRETE_ALM_AFB looks like its APACS+ counterpart. The DI input was changed to DIN for compatibility with the HMI.

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
ALARM	BOOL	Alarm 1	False	False	

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
LIM	BOOL	Alarm Limit	False	True	
DELAYIN	TIME	Into Alarm Delay Time	T#0d0h0m0s0ms	False	
DELAYOUT	TIME	Out of Alarm Delay Time	T#0d0h0m0s0ms	False	
DISABLE_ON_OOS	BOOL	Disable on Out of Service	False	False	

2.10 EXT_SET_AFB

2.10.1 EXT_SET: External Setpoint Controller

Function Block Name (compiled version of the Chart)	EXT_SET_AFB
Function Block Number	FB605
Chart Name	EXT_SET
See also, Template Chart (usage example)	**EXT_SET_AT**

Function

The External Setpoint Controller (EXT_SET_AFB) block is a factory-compiled function block. It is intended for use as an external-set PID controller with tracking setpoint. High and low alarms are provided for the process variable. Another alarm is provided to detect absolute deviation between the process variable and setpoint.

EXT_SET_AFB is a compiled version of the EXT_SET source chart. Within the chart, there is a control loop that contains SETPNT_AFB, PID_AFB, and AM_AFB blocks along with alarming logic.

The SETPOINT block supplies the setpoint [SP_OUT] to the CONTROLLER block. The SETPOINT block has a ramping feature for internal mode (EXT=FALSE). In external mode (EXT=TRUE), the external setpoint [ESP] becomes the controller setpoint, and the SETPOINT block's internal setpoint (ISP) tracks the external setpoint, thereby ensuring no change in setpoint during transition to internal mode. The setpoint tracks the process variable [PV] in manual mode (AUTO=FALSE).

The CONTROLLER block is a PID block. It operates in auto mode (AUTO=TRUE) and tracks the downstream output [OUT] in manual mode (AUTO=FALSE). The AUTO_MANUAL block provides manual control of the output. In auto mode the CONTROLLER block output becomes the loop output [OUT]. In manual mode the AUTO_MANUAL block internal output IOUT becomes the loop output.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32). Additionally, it is called in OB100 and in OB102 (see Start-up Characteristics).

Start-up Characteristics

On a warm restart, OB100 is called and parameters retain the values they had immediately before the restart. On a cold restart, OB102 is called and parameters are initialized with the last downloaded values.

The warm and cold startup behavior of AUTO, SP, EXT, and OUT is configurable, allowing the user to override the default PCS 7 behavior described above if necessary. As delivered, AUTO will be forced to "Manual"EXT will be forced to

"Internal" and OUT will be forced to 0.0 on a CPU restart. SP will be set in accordance with default PCS 7 behavior.

To force the AUTO parameter to a specified value on a restart, set the RSTRT_AUTO_FLAG to true ("Force AUTO") and configure RSTRT_AUTO_VALUE with the desired forced value ("Auto" or "Manual"). The RSTRT_AUTO_VALUE will be used to initialize AUTO, if and only if the RSTRT_AUTO_FLAG is true. If RSTRT_AUTO_FLAG is set to false ("Default"), the AUTO parameter will be set in accordance with default PCS 7 behavior.

To force the SP parameter to a specified value on a restart, set the RSTRT_SP_FLAG to true ("Force SP") and configure RSTRT_SP_VALUE with the desired forced value. The RSTRT_SP_VALUE will be used to initialize SP, if and only if the RSTRT_SP_FLAG is true and the block restarts in Auto mode (AUTO="Auto"). If RSTRT_SP_FLAG is set to false ("Default"), the SP parameter will be set in accordance with default PCS 7 behavior.

To force the EXT parameter to a specified value on a restart, set the RSTRT_EXT_FLAG to true ("Force EXT") and configure RSTRT_EXT_VALUE with the desired forced value ("Internal or External"). The RSTRT_EXT_VALUE will be used to initialize EXT, if and only if the RSTRT_EXT_FLAG is true. If RSTRT_EXT_FLAG is set to false ("Default"), the EXT parameter will be set in accordance with default PCS 7 behavior. Forcing the EXT parameter to "External" will cause the SP to be set to the ESP, thereby overriding the SP restart parameter settings.

To force the OUT parameter to a specified value on a restart, set the RSTRT_OUT_FLAG to true ("Force OUT") and configure RSTRT_OUT_VALUE with the desired forced value. The RSTRT_OUT_VALUE will be used to initialize OUT, if and only if the RSTRT_OUT_FLAG is true and the block restarts in Manual mode (AUTO = "Manual"). If RSTRT_OUT_FLAG is set to false ("Default"), the OUT parameter will be set in accordance with default PCS 7 behavior.

Warning

Using "Default" for the RSTRT_OUT_FLAG could result in your output device retaining its last operating state after a restart. Similarly, using "Default" for the RSTRT_AUTO_FLAG could result in your process running in automatic mode after a restart. Use "Forced" for the restart flags and configure the associated restart values to ensure a known restart state.

If restart parameter values are changed offline, the configuration must be compiled (changes only) and downloaded (changes only) to the controller in order for the settings to take effect upon restart. If restart parameter values are changed online (via CFC Test mode), the configuration must be compiled (entire program) and downloaded (entire program) to the controller in order for the settings to take effect upon cold restart. This step is not necessary for the online changes to take effect upon a warm restart.

Operator Control and Monitoring (M&C Relevant)

See Faceplate Information

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

ALARM_8P ID	Message #	Message Class	Default Message Text
EV_ID	1	Warning - above	HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID	2	Warning - below	LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID	3	Alarm - above	HIGH_DEV - $$BlockComment$$ - Limit = @6%f@
EV_ID	4	Alarm - below	LOW_DEV - $$BlockComment$$ - Limit = @7%f@
EV_ID	5	Alarm - above	ABSOLUTE_DEV - $$BlockComment$$ - Limit = @8%f@
EV_ID	6	PLC Process Control Message - Failure	OUT_OF_RANGE - $$BlockComment$$
EV_ID	7	PLC Process Control Message - Failure	QUALITY_NOT_GOOD - $$BlockComment$$
EV_ID2	1	Alarm - above	HIGH_ HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID2	2	Alarm - below	LOW_ LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID2	3	Warning - above	INC_ROC - $$BlockComment$$ - Limit = @6%f@
EV_ID2	4	Warning - below	DEC_ROC - $$BlockComment$$ - Limit = @7%f@
EV_ID2	5	Warning - above	ABS_ROC - $$BlockComment$$ - Limit = @8%f@

Assignment of auxiliary process values to the block parameters

ALARM_8P ID	Value	Block Parameter
EV_ID	@1%s@	BA_NA
EV_ID	@2%d@	BA_ID
EV_ID	@3%d@	STEP_NO
EV_ID	@4%f@	H_LIM
EV_ID	@5%f@	L_LIM
EV_ID	@6%f@	HD_LIM
EV_ID	@7%f@	LD_LIM

ALARM_8P ID	Value	Block Parameter
EV_ID	@8%f@	AD_LIM
EV_ID2	@1%s@	BA_NA
EV_ID2	@2%d@	BA_ID
EV_ID2	@3%d@	STEP_NO
EV_ID2	@4%f@	HH_LIM
EV_ID2	@5%f@	LL_LIM
EV_ID2	@6%f@	IR_LIM
EV_ID2	@7%f@	DR_LIM
EV_ID2	@8%f@	AR_LIM

2.10.1.1 EXT_SET_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
PV	REAL	Process Variable	0	True	
TSP	REAL	SP target value for SP ramping	0	True	
H_EN	BOOL	Enable High Alarm	True	True	
L_EN	BOOL	Enable Low Alarm	True	True	
HD_EN	BOOL	Enable High Deviation Alarm	False	True	
LD_EN	BOOL	Enable Low Deviation Alarm	False	True	
AD_EN	BOOL	Enable Absolute Deviation Alarm	True	True	
OOR_EN	BOOL	Enable Out of Range Alarm	False	True	
QUAL_EN	BOOL	Enable Quality Not Good Alarm	False	True	
HH_EN	BOOL	Enable High High Alarm	False	True	
LL_EN	BOOL	Enable Low Low Alarm	False	True	
IR_EN	BOOL	Enable Increasing Rate of Change Alarm	False	True	
DR_EN	BOOL	Enable Decreasing Rate of Change Alarm	False	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
AR_EN	BOOL	Enable Absolute Rate of Change Alarm	False	True	
OOS	BOOL	Out of Service indicator	False	True	
QBAD	BOOL	Bad quality indicator	False	False	
EV_ID	DWORD	Event ID	Compiler generated for the type or N/A if used as chart *	False	
EV_ID2	DWORD	Event ID	Compiler generated for the type or N/A if used as chart *	False	
ESP	REAL	External set point	0	True	
SAMPLE_T	REAL	Current scan rate supplied by OS	Compiler generated for the type or N/A if used as chart *	False	
RSTRT_AUTO_FLAG	BOOL	Default: Retain on WS, Use load value on CS. Force: Use RSTRT_AUTO_ VALUE on restart	True	False	
RSTRT_AUTO_VALUE	BOOL	Init restart value for Auto Mode if RSTRT_AUTO_ FLAG true	False	False	
RSTRT_EXT_FLAG	BOOL	Default: Retain on WS, Use load value on CS. Force: Use RSTRT_EXT_V ALUE on restart	True	False	
RSTRT_EXT_VALUE	BOOL	Init restart value for EXTernal if RSTRT_EXT_F LAG true	False	False	
RSTRT_SP_FLAG	BOOL	Default: Retain on WS, Use load value on CS. Force: Use RSTRT_SP_VA LUE on restart.	False	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
RSTRT_SP_VALUE	REAL	Init restart value for SP if RSTRT_SP_FLAG true and Auto mode.	0	False	
RSTRT_OUT_FLAG	BOOL	Default: Retain on WS, Use load value on CS. Force: Use RSTRT_OUT_VALUE on restart.	True	False	
RSTRT_OUT_VALUE	REAL	Init restart value for OUT if RSTRT_OUT_FLAG true and manual mode	0	False	
BA_EN	BOOL	Batch enabled	False	True	
OCCUPIED	BOOL	Batch occupied	False	True	
BA_ID	DWORD	Batch ID number	0	True	
BA_NA	STRING[32]	Batch name	‘‘	True	
STEP_NO	DWORD	Batch step number	0	True	
DecimalPlacesPV	INT	Decimal places displayed in HMI for PV, SP etc.	2	True	
DecimalPlacesOUT	INT	Decimal places displayed in HMI for OUT, OUT Limits	2	True	
ConfirmChange	BOOL	If TRUE changes from HMI must be confirmed	False	True	
LIOP_MAN_SEL	BOOL	Select AUTO/MANUAL: 1 = Linking, 0 = Operator active	0	False	
AUT_L	BOOL	Linkable input for AUTO/MANUAL mode, 1 = AUTO	0	False	
LIOP_EXT_SEL	BOOL	Select External/Internal SP flag: 1 = from Link, 0 = from Operator	0	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
EXTRNL_L	BOOL	Linkable input to select External/Internal SP flag, 1=Extrnl	0	False	
LIOP_ ISP_SEL	BOOL	Select internal SP: 1 = from Link, 0 = from Operator	0	False	
SP_L	REAL	Linkable input for internal Setpoint	0.0	False	
LIOP_OUT_SEL	BOOL	Select writing to OUT: 1 = from Link, 0 = from Operator	0	False	
OUT_L	REAL	Linkable input to be written to OUT	0.0	False	

Note

This list differs significantly from the inputs of the EXT_SET_AFB block in APACS+, since the APACS+ list includes only the PV input and ESP (actually named EXT_SP in APACS+) input.

*The Update Sampling Time option must be checked when compiling a chart that contains the EXT_SET_AFB for these values to be automatically generated by the compiler.

Input parameters added for PCS 7 V6.1 include:

- Batch parameters BA_EN, OCCUPIED are added to support future batch faceplate view.
- Batch parameters BA_NA, BA_ID, STEP_NO are added to pass batch values in alarm messages.
- DecimalPlacesPV, DecimalPlacesOUT added to control the number of displayed decimal places in HMI.
- ConfirmChange added to give APACS users expected feel of "one-step" operator entry.
- LIOP_MAN_SEL and AUT_L provide a linkable connection for the program to set AUTO, without using SET_VAL.
- LIOP_ISP_SEL and SP_L provide a linkable connection for the program to set internal Setpoint, without using SET_VAL.
- LIOP_OUT_SEL and OUT_L provide a linkable connection for the program to set OUT, without using SET_VAL.
- LIOP_EXT_SEL and EXTRNL_L provide a linkable connection for the program to set External Setpoint mode, without using SET_VAL.

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
SP_OUT	REAL	Calculated set point value	0	False	
OUT	REAL	Calculated output variable that has been limited	0	False	
ALARM	BOOL	Alarm Status	False	False	
H_ALM	BOOL	When TRUE, the High alarm is detected and enabled.	False	False	
L_ALM	BOOL	When TRUE, the Low alarm is detected and enabled.	False	False	
HD_ALM	BOOL	When TRUE, the High Deviation alarm is detected and enabled.	False	False	
LD_ALM	BOOL	When TRUE, the Low Deviation alarm is detected and enabled.	False	False	
AD_ALM	BOOL	When TRUE, the Absolute Deviation alarm is detected and enabled.	False	False	
OOR_ALM	BOOL	When TRUE, the Out of Range alarm is detected and enabled.	False	False	
QUAL_ALM	BOOL	When TRUE, the Quality Not Good alarm is detected and enabled.	False	False	
HH_ALM	BOOL	When TRUE, the High High alarm is detected and enabled.	False	False	
LL_ALM	BOOL	When TRUE, the Low Low alarm is detected and enabled.	False	False	
IR_ALM	BOOL	When TRUE, the Increasing Rate of Change alarm is detected and enabled.	False	False	
DR_ALM	BOOL	When TRUE, the Decreasing Rate of Change alarm is detected and enabled.	False	False	
AR_ALM	BOOL	When TRUE, the Absolute Rate of Change alarm is detected and enabled.	False	False	
VL_EUMIN_OUT	REAL	Copy of Engineering unit: Output, Low	0	False	
VL_EUMAX_OUT	REAL	Copy of Engineering unit: Output, High	0	False	

Note

The APACS+ SP output was renamed to SP_OUT to avoid conflict with the newly added SP In/Out parameter. SP_OUT is used to interconnect the set-point value to other entities on a chart. The OS interaction is handled by the SP InOut parameter.

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
AUTO	BOOL	Auto/manual mode control	False	True	
EXT	BOOL	External/internal mode control	False	True	

Name	Data Type	Explanation	Initial	OC& M	Valid Values
SP	REAL	Computer set point value	0	True	
RAMP	BOOL	SP ramping on/off mode control	False	True	
R_TIME	TIME	Ramp time	T#0MS	True	0 to 24d20h31m23s647ms
SPL	REAL	Set point low limit	0	True	
SPH	REAL	Set point high limit	100	True	
R_RATE	REAL	Ramp rate (in engineering units per min)	0	True	
USE_RR	BOOL	Ramp mode control. Use RTIME or RRATE to control ramp	False	True	
PV_TRACK	BOOL	Internal setpoint tracking control. When Track, internal setpoint tracks PV	True	True	
PG	REAL	Proportional Gain	0.001	True	0.001 to 100.0
TI	REAL	Time-Integral Minutes per Reset	4000	True	0.0 to 4000.0
TD	REAL	Time-Derivative Minutes	0	True	0.00 to 100.0
DG	REAL	Derivative Gain	10	True	2.00 to 20.00
DIR	BOOL	Reverse or Direct action	False	True	
ADAPTIVE_PG_INIT	BOOL	Adaptive Gain Init flag	False	True	
PV_EUMIN	REAL	Engineering unit: Input, Low	0	True	
PV_EUMAX	REAL	Engineering unit: Input, High	100	True	
VL_EUMIN	REAL	Engineering unit: Output, Low	0	True	
VL_EUMAX	REAL	Engineering unit: Output, High	100	True	
VL	REAL	Computer Output, typically written to by an operator command	0	True	
VLL	REAL	OUT low limit input	-3.3	True	
VLH	REAL	OUT high limit input	103.3	True	
H_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the High alarm.	100	True	
H_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the High alarm to clear.	0.5	True	
H_DELAYIN	TIME	The amount of time the input must reach or exceed the High limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
H_DELAYOUT	TIME	The amount of time the input must be within the High limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
L_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the Low alarm.	0	True	
L_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Low alarm to clear.	0.5	True	

Name	Data Type	Explanation	Initial	OC& M	Valid Values
L_DELAYIN	TIME	The amount of time the input must reach or exceed the Low limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
L_DELAYOUT	TIME	The amount of time the input must be within the Low limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
HD_LIM	REAL	The value (in engineering units) to which (PV) – (DEV) is compared for the High Deviation alarm.	0	True	
HD_DB	REAL	The amount of engineering units by which (PV) – (DEV) must be within the limit for the High Deviation alarm to clear.	0.5	True	
HD_DELAYIN	TIME	The amount of time the input must reach or exceed the High Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
HD_DELAYOUT	TIME	The amount of time the input must be within the High Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
LD_LIM	REAL	The value (in engineering units) to which (DEV) - (PV) is compared for the Low Deviation alarm.	0	True	
LD_DB	REAL	The amount of engineering units by which (DEV) - (PV) must be within the limit for the Low Deviation alarm to clear.	0.5	True	
LD_DELAYIN	TIME	The amount of time the input must reach or exceed the Low Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
LD_DELAYOUT	TIME	The amount of time the input must be within the Low Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
AD_LIM	REAL	The value (in engineering units) to which ABS((PV) – (DEV)) is compared for the Absolute Deviation alarm.	100	True	
AD_DB	REAL	The amount of engineering units by which ABS((PV) – (DEV)) must be within the limit for the Absolute Deviation alarm to clear.	0.5	True	
AD_DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms

Name	Data Type	Explanation	Initial	OC& M	Valid Values
AD_DELAYOUT	TIME	The amount of time the input must be within the Absolute Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
HH_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the High High alarm.	0	True	
HH_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the High High alarm to clear.	0.5	True	
HH_DELAYIN	TIME	The amount of time the input must reach or exceed the High High limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
HH_DELAYOUT	TIME	The amount of time the input must be within the High High limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
LL_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the Low Low alarm.	0	True	
LL_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Low Low alarm to clear.	0.5	True	
LL_DELAYIN	TIME	The amount of time the input must reach or exceed the Low Low limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
LL_DELAYOUT	TIME	The amount of time the input must be within the Low Low limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
IR_LIM	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Increasing Rate of Change alarm.	0	True	
IR_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Increasing Rate of Change alarm to clear.	0.5	True	
IR_DELAYIN	TIME	The amount of time the input must reach or exceed the Increasing Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
IR_DELAYOUT	TIME	The amount of time the input must be within the Increasing Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms

Name	Data Type	Explanation	Initial	OC&M	Valid Values
DR_LIM	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Decreasing Rate of Change alarm.	0	True	
DR_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Decreasing Rate of Change alarm to clear.	0.5	True	
DR_DELAYIN	TIME	The amount of time the input must reach or exceed the Decreasing Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
DR_DELAYOUT	TIME	The amount of time the input must be within the Decreasing Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
AR_LIM	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Absolute Rate of Change alarm.	0	True	
AR_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Absolute Rate of Change alarm to clear.	0.5	True	
AR_DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
AR_DELAYOUT	TIME	The amount of time the input must be within the Absolute Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
OOR_DELAYIN	TIME	The amount of time the input must be outside the limits EINH, EINL to trip the Out of Range alarm.	T#0MS	False	0 to 24d20h31m23s647ms
OOR_DELAYOUT	TIME	The amount of time the input must be within the limits EINH, EINL to cause the Out of Range alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
QUAL_DELAYIN	TIME	The amount of time the input quality must be bad to trip the Quality Not Good alarm.	T#0MS	False	0 to 24d20h31m23s647ms
QUAL_DELAYOUT	TIME	The amount of time the input quality must be good to cause the Quality Not Good alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
DISABLE_ON_OOS	BOOL	When set to TRUE, the block is disabled if (OOS) is also TRUE.	False	False	

2.11 MOTOR_A_AFB

2.11.1 MOTOR_A_AFB: Standard Motor

Function Block Name	MOTOR_A_AFB
Function Block Number	FB505

Function

The Standard Motor Function Block (MOTOR_A_AFB) provides standard motor start/stop functions that can be easily interfaced to an operator station. This block is based on the APACS+ MOTOR_A function block. This block can receive commands from programmed logic, such as a sequential function chart (SFC) while in automatic mode, or it can receive operator-entered commands from an operator interface while in manual mode.

The PROOF input receives the state of the motor (TRUE = running and FALSE = stopped). The OUT output sets the state of the motor (TRUE = running and FALSE = stop).

The AUTO parameter indicates the auto/manual state. This state can be changed external to the block. It can also be changed by the block itself, depending on In/Out parameters. When the block is in manual mode (AUTO = FALSE), commands to the block are received via a write of TRUE to the START or STOP In/Out parameters. The block automatically resets these values after receiving the command. When the block is in auto mode (AUTO = TRUE), commands to the block are received via the A_STRT and A_STOP inputs. Since auto mode is intended for use with commands sent by an SFC, there are some In/Out settings that accommodate this use:

- When AUTO_TRACK is TRUE, the AUTO mode tracks the MODE input (intended to be the mode of the SFC), such that AUTO is set TRUE when MODE becomes TRUE.
- If MODE_HOLD_AUTO is TRUE, AUTO is held TRUE as long as MODE is TRUE. When FALSE, the AUTO value is set when the MODE input value is TRUE, but can then be manually changed.
- If ACCEPT_ANY_AUTO is TRUE, AUTO is set TRUE whenever the A_STRT (Automatic Start) or A_STOP (Automatic Stop) input changes state.

When in auto mode, the positive transition of A_STOP (POS_TRANS_ASTOP) value determines the function of the A_STRT and A_STOP inputs. If the POS_TRANS_ASTOP is TRUE, the commands need to be FALSE-to-TRUE transitions for both the A_STRT and A_STOP inputs. Unlike the START and STOP command values, the A_STRT and A_STOP input values must be reset external to the block. If POS_TRANS_ASTOP value is FALSE, the same value should be placed on both the A_STRT and A_STOP inputs so that a FALSE-to-TRUE transition sends a start command and a TRUE-to-FALSE transition sends a stop command. To avoid the need to generate such a transition for these commands when POS_TRANS_ASTOP is FALSE, the AUTO_TRANS_REQ value can be set FALSE. The A_STRT input (with the same value on the A_STOP input) is used as the desired state of the motor (TRUE = running, FALSE = stop).

There are many permissive/interlock features built into this block. The PERMIT input, if configured, must be TRUE to permit any start command to take effect. The block has 11 interlock inputs (INT01 ... INT11), but only the first two are visible by default. All of the configured interlock inputs must be TRUE for the motor to start or remain running; the INTRLK (Interlock) output value indicates whether all configured interlock inputs are TRUE. The OVERRD (Override) value represents the override status: when TRUE, any FALSE interlocks (or permit) are overridden, the INTRLK value is set TRUE, and the motor is allowed to run. OVERRD may be enabled by a tag write from outside the block. For display at an HMI, there are In/Out values that hold descriptors for the permit (PERMIT) and the interlock (INT01...INT11) inputs.

This block contains failure-checking features that set the FAIL value to TRUE when a problem is detected. The detected problems include:

- Failure to start before the STRT_T has expired
- Failure to stop before the STOP_T has expired
- Loss of proof
- Uncommanded stop due to loss of an interlock (enabled by setting the FAIL_ON_INTRLK value to TRUE)
- Rejected start command due to missing permissive or interlock (checking for this failure is always enabled when MODE = TRUE and for all conditions by setting the FAIL_ON_STRT_CMD value to TRUE.)

The FAIL value is set TRUE when a fail condition is detected. It remains TRUE until it is reset by a tag write from outside of the block (by acknowledge logic or an operator command). When FAIL is TRUE, the OUT output value is held FALSE. The ALARM_STATE value, an integer, is provided to indicate the specific failure condition that occurred.

The READY output indicates that the motor is ready to receive auto commands, where TRUE indicates that the motor is in auto mode and the interlocks are not overridden. The STATE output is an integer value that represents the various states of the motor.

The MOTOR_A_AFB supports alarming. An alarm message will be generated if the FAIL parameter becomes TRUE. In APACS+, this functionality was not part of the MOTOR_A block, but was instead handled by ProcessSuite.

Interlock Configuration

The Interlock parameters are not extensible as they were in *4-mation*. In the PCS 7 implementation of this block, all of the Interlocks exist with only the first two visible by default. If more than two interlocks are required, each additional interlock must be made visible.

For each required interlock, configure the following:

- Connect a boolean interlock condition to the Interlock input.
- Configure the description so that it is no longer null ('). If the description is left as a null, the faceplate will not display the status for this interlock. The faceplate is designed to hide an interlock condition when the interlock description is null.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32). Additionally, it is called in OB100 and in OB102 (see Start-up Characteristics).

Start-up Characteristics

On a warm restart, OB100 is called and parameters retain the values they had immediately before the restart. On a cold restart, OB102 is called and parameters are initialized with the last downloaded values.

The warm and cold startup behavior of AUTO and OUT is configurable, allowing the user to override the default PCS 7 behavior described above if necessary. As delivered, AUTO will be forced to "Manual" and OUT will be forced to "False" on a CPU restart.

To force the AUTO parameter to a specified value on a restart, set the RSTRT_AUTO_FLAG to true ("Force AUTO") and configure RSTRT_AUTO_VALUE with the desired forced value ("Auto" or "Manual"). The RSTRT_AUTO_VALUE will be used to initialize AUTO, if and only if the RSTRT_AUTO_FLAG is true. If RSTRT_AUTO_FLAG is set to false ("Default"), the AUTO parameter will be set in accordance with default PCS 7 behavior.

To force the OUT parameter to a specified value on a restart, set the RSTRT_OUT_FLAG to true ("Force OUT") and configure RSTRT_OUT_VALUE with the desired forced value ("True" or "False"). The RSTRT_OUT_VALUE will be used to initialize OUT, if and only if the RSTRT_OUT_FLAG is true. If RSTRT_OUT_FLAG is set to false ("Default"), the OUT parameter will be set in accordance with default PCS 7 behavior.

Warning

Using "Default" for the RSTRT_OUT_FLAG could result in your output device retaining its last operating state after a restart. Similarly, using "Default" for the RSTRT_AUTO_FLAG could result in your process running in automatic mode after a restart. Use "Forced" for the restart flags and configure the associated restart values to ensure a known restart state.

If restart parameter values are changed offline, the configuration must be compiled (changes only) and downloaded (changes only) to the controller in order for the settings to take effect upon restart. If restart parameter values are changed online (via CFC Test mode), the configuration must be compiled (entire program) and downloaded (entire program) to the controller in order for the settings to take effect upon cold restart. This step is not necessary for the online changes to take effect upon a warm restart.

Operator Control and Monitoring (M&C Relevant)

See Faceplate Information

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

Message #	Message Class	Default Message Text
1	PLC Process Control Message - Failure	Motor Feedback Failure - $$BlockComment$$

Assignment of auxiliary process values to the block parameters

Value	Block Parameter
@1%s@	BA_NA
@2%d@	BA_ID
@3%d@	STEP_NO

2.11.1.1 MOTOR_A_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC& M	Valid Values
PROOF	BOOL	Running status feedback signal from the motor	False	True	
A_STRT	BOOL	Automatic Start command input	False	False	
A_STOP	BOOL	Automatic Stop command input	False	False	
PERMIT	BOOL	Permitted to Start input	True	True	
MODE	BOOL	Mode input from of a sequential function chart. 0=Manual, 1=Auto	False	False	
INT01	BOOL	Interlock 1	True	True	
INT02	BOOL	Interlock 2	True	True	
INT03	BOOL	Interlock 3	True	True	
INT04	BOOL	Interlock 4	True	True	
INT05	BOOL	Interlock 5	True	True	
INT06	BOOL	Interlock 6	True	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
INT07	BOOL	Interlock 7	True	True	
INT08	BOOL	Interlock 8	True	True	
INT09	BOOL	Interlock 9	True	True	
INT10	BOOL	Interlock 10	True	True	
INT11	BOOL	Interlock 11	True	True	
RSTRT_AUTO_FLAG	BOOL	Default: Retain on WS, Use load val on CS. Force: Use RSTRT_AUTO_VALUE on restart	True	False	
RSTRT_AUTO_VALUE	BOOL	Init restart value for Auto Mode if RSTRT_AUTO_FLAG true	False	False	
RSTRT_OUT_FLAG	BOOL	Default: Retain on WS, Use load val on CS. Force: Use RSTRT_OUT_VALUE on restart	True	False	
RSTRT_OUT_VALUE	BOOL	Init restart value for OUT if RSTRT_OUT_FLAG	False	False	
PERMIT_DESC	STRING[32]	Permit Description	''	True	
INT01_DESC	STRING[32]	Interlock 1 Description	''	True	
INT02_DESC	STRING[32]	Interlock 2 Description	''	True	
INT03_DESC	STRING[32]	Interlock 3 Description	''	True	
INT04_DESC	STRING[32]	Interlock 4 Description	''	True	
INT05_DESC	STRING[32]	Interlock 5 Description	''	True	
INT06_DESC	STRING[32]	Interlock 6 Description	''	True	
INT07_DESC	STRING[32]	Interlock 7 Description	''	True	
INT08_DESC	STRING[32]	Interlock 8 Description	''	True	
INT09_DESC	STRING[32]	Interlock 9 Description	''	True	
INT10_DESC	STRING[32]	Interlock 10 Description	''	True	
INT11_DESC	STRING[32]	Interlock 11 Description	''	True	
EV_ID	DWORD	Event ID	(Compiler generated)	False	
LIOP_MAN_SEL	BOOL	Select AUTO/MANUAL: 1 = Linking, 0 = Operator active	False	False	
AUT_L	BOOL	Linkable input for AUTO/MANUAL mode, 1 = AUTO	False	False	
BA_EN	BOOL	Batch enabled	False	True	
OCCUPIED	BOOL	Batch occupied	False	True	
BA_ID	DWORD	Batch ID number	0	True	

Name	Data Type	Explanation	Initial	OC& M	Valid Values
STEP_NO	DWORD	Batch step number	0	True	
BA_NA	STRING[32]	Batch name	‘‘	True	
ConfirmChange	BOOL	If TRUE changes from HMI must be confirmed	False	True	

Note

This list differs from the inputs of the MOTOR_A_AFB block in APACS+ as follows:

- Each input shown above with datatype STRING was a softlist parameter in APACS+.
- The INT01_DESC through INT09_DESC have slightly different names than their APACS+ counterparts. In APACS+ a leading zero was excluded, making the names INT1_DESC through INT9_DESC.
- EV_ID parameter is a new input and is required for the PCS 7 version to support alarming. This parameter did not exist in APACS+.
- Restart parameters (RSTRT_...) did not exist in APACS+.
- LIOP_MAN_SEL and AUT_L added to provide alternate for SET_VAL function.
- Batch parameters BA_EN, OCCUPIED are added to support future batch faceplate view.
- Batch parameters BA_NA, BA_ID, STEP_NO are added to pass batch values in alarm messages.
- ConfirmChange added to give APACS+ users expected feel of "one-step" operator entry

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
OUT	BOOL	The block’s Output that is used to ultimately control the motor.	False	True	
ALARM	BOOL	Indicates when the motor is in alarm.	False	False	
INTRLK	BOOL	Interlock OK output	False	True	
READY	BOOL	Ready output	False	False	
STATE	INT	State of the motor	0	False	
ALARM_STATE	INT	Alarm state	0	False	

Note

This list differs from the outputs of the MOTOR_A_AFB block in APACS+ as follows:

- FAIL and AUTO and OVERRD have been moved to In/Out in PCS 7.
- ALARM_STATE is a new output. In APACS+, it was a softlist parameter.
- ALARM is a new output and is present to support the alarming functionality. This output along with the alarm functionality did not exist in APACS+.
- INTRLK nub is named INT_OK in APACS+. The ProcessSuite name "INTRLK" was preferred.

Integer Values for Motor State

VALUE	STATE	FAIL	OVERRD	PROOF	OUT	INTRLK & PERMIT
0	Undefined	-	-	-	-	-
1	Transition	False	False	< >OUT	< >PROOF	True
2	Stopped	False	False	False	False	True
3	Running	False	False	True	True	-
4	Stopped and Fail	True	-	False	-	-
5	Running and Fail	True	-	True	-	-
6	Stopped and Override	False	True	False	-	-
7	Running and Override	False	True	True	-	-
8	Stopped and Interlock not OK	False	False	False	False	False

"<>" means "not equal to"

"-" means "don't care"

Integer Values for Motor Alarm (Failure) State

Value	State	Status
0	OK	No alarm
1	Failed to Start	PROOF value not reset in time after OUT value was set TRUE
2	Failed to Stop	PROOF value went FALSE while OUT value remained TRUE
3	Loss of PROOF	PROOF value went FALSE while OUT value remained TRUE
4	Inappropriate PROOF	PROOF value went TRUE while OUT value remained FALSE
5	Uncommanded Stop	INTRLK value went FALSE while OUT value was TRUE, causing OUT to be SET FALSE
6	Rejected Start Command	INTRLK or PERMIT value was FALSE when a start command was issued

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
FAIL	BOOL	Indicates whether motor has fault condition	False	True	
AUTO	BOOL	Auto mode output	False	True	
OVERRD	BOOL	Override output	False	True	
AUTO_TRACK	BOOL	If true, block is set to auto mode when SFC transitions to AUTO (MODE becomes "True").	False	False	
MODE_HOLD_AUTO	BOOL	Indicates whether block held in auto mode while MODE is "True"	False	False	
ACCEPT_ANY_AUTO	BOOL	When true, block is set to auto when A_STRT or A_STOP changes state	False	False	
POS_TRANS_ASTOP	BOOL	Indicates whether a positive transition is required to stop the motor	True	False	
AUTO_TRANS_REQ	BOOL	When true the A_STRT and A_STOP inputs need to change state to issue start/stop	True	False	
FAIL_ON_INTRLK	BOOL	Fail on interlock. When TRUE and the loss of an interlock stops the motor (ie. Uncommanded stop), the FAIL status is set TRUE, and the ALARM_STATE parameter is set to its appropriate valve.	False	False	
FAIL_ON_STRT_CMD	BOOL	Fail on start command. When TRUE and a start command is rejected due to the interlocks of permissive, the FAIL status is set TRUE and the ALARM_STATE is set to its appropriate value.	False	False	
START	BOOL	Start command value for manual mode	False	True	
STOP	BOOL	Stop command value for manual mode	False	True	
STRT_T	TIME	Maximum time allowed to start the motor	T#10S	True	
STOP_T	TIME	Maximum time allowed to stop the motor	T#10S	True	
A1EN	BOOL	Alarm 1 Enable	True	True	
A1LIM	BOOL	Alarm 1 Limit	True	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
A1DELAYIN	TIME	Into Alarm 1 Delay Time	T#0MS	False	0 to 24d20h31m 23s647ms
A1DELAYOUT	TIME	Out of Alarm 1 Delay Time	T#0MS	False	0 to 24d20h31m 23s647ms
OOS	BOOL	Out of Service indicator	False	True	
DISABLE_ON_OOS	BOOL	Disable on Out of Service	False	False	

Note

This list differs from the softlist parameters of the MOTOR_A_AFB block in APACS+ as follows:

- FAIL, AUTO, and OVERRD which were outputs in APACS+, have been moved to In/Out in PCS 7. This is necessary to allow writes from the HMI.
- ALARM_STATE was moved to Output.
- The PT_DESC parameter has been eliminated. Instead, the Comment property of the block is being used to contain the point description information. This is because the faceplate will automatically display the information provided in the Comment field..
- INT01_DESC through INT10_DESC and PERMIT_DESC have been moved to Inputs.
- STRT_T and STOP_T are named START_T and STOP_T in APACS+. The ProcessSuite names STRT_T and STOP_T are preferred.
- The following alarm parameters are new: A1EN, A1LIM, A1DELAYIN, A1DELAYOUT, OOS, DISABLE_ON_OOS. These are needed to support the alarming functionality.

2.12 MOTOR_A_ALM_AFB

2.12.1 MOTOR_A_ALM: Motor with Alarm

Function Block Name (compiled version of the Chart)	MOTOR_A_ALM_AFB
Function Block Number	FB613
Chart Name	MOTOR_A_ALM
See also, Template Chart (usage example)	**MOTOR_A_ALM_AT**

Function

This block provides standard motor start/stop equipment module functions (similar to the MOTOR_1OUT). Device alarming functions are provided by this block.

This factory-compiled block is an encapsulation of the MOTOR_A_AFB block with a discrete alarm block. To provide thorough alarm reporting, the alarm state value of the MOTOR_A_AFB block is used to create separate alarms for the following conditions:

- Failure to start (or loss of proof)
- Failure to stop (or unexpected proof)
- Uncommanded stop or rejected start command (due to interlock)

The latched fail state of the MOTOR_A_AFB block is unlatched automatically when the alarm is acknowledged.

The parameters are generally identically to those of the MOTOR_A_AFB block. See MOTOR_A_AFB for more details and a complete description of the functionality of this block.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32). Additionally, it is called in OB100 and in OB102 (see Start-up Characteristics).

Start-up Characteristics

On a warm restart, OB100 is called and parameters retain the values they had immediately before the restart. On a cold restart, OB102 is called and parameters are initialized with the last downloaded values.

The warm and cold startup behavior of AUTO and OUT is configurable, allowing the user to override the default PCS 7 behavior described above if necessary. As delivered, AUTO will be forced to "Manual" and OUT will be forced to "False" on a CPU restart.

To force the AUTO parameter to a specified value on a restart, set the RSTRT_AUTO_FLAG to true ("Force AUTO") and configure

RSTRT_AUTO_VALUE with the desired forced value ("Auto" or "Manual"). The RSTRT_AUTO_VALUE will be used to initialize AUTO, if and only if the RSTRT_AUTO_FLAG is true. If RSTRT_AUTO_FLAG is set to false ("Default"), the AUTO parameter will be set in accordance with default PCS 7 behavior.

To force the OUT parameter to a specified value on a restart, set the RSTRT_OUT_FLAG to true ("Force OUT") and configure RSTRT_OUT_VALUE with the desired forced value ("True" or "False"). The RSTRT_OUT_VALUE will be used to initialize OUT, if and only if the RSTRT_OUT_FLAG is true. If RSTRT_OUT_FLAG is set to false ("Default"), the OUT parameter will be set in accordance with default PCS 7 behavior.

Warning

Using "Default" for the RSTRT_OUT_FLAG could result in your output device retaining its last operating state after a restart. Similarly, using "Default" for the RSTRT_AUTO_FLAG could result in your process running in automatic mode after a restart. Use "Forced" for the restart flags and configure the associated restart values to ensure a known restart state.

If restart parameter values are changed offline, the configuration must be compiled (changes only) and downloaded (changes only) to the controller in order for the settings to take effect upon restart. If restart parameter values are changed online (via CFC Test mode), the configuration must be compiled (entire program) and downloaded (entire program) to the controller in order for the settings to take effect upon cold restart. This step is not necessary for the online changes to take effect upon a warm restart.

Interlock Configuration

All of the Interlocks for this block are visible by default. For each required interlock, configure the following:

- Connect a boolean interlock condition to the Interlock input.
- Configure the description so that it is no longer null ('). If the description is left as a null, the faceplate will not display the status for this interlock. The faceplate is designed to hide an interlock condition when the interlock description is null.

Operator Control and Monitoring (M&C Relevant)

See Faceplate Information

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

Message #	Message Class	Default Message Text
1	PLC Process Control Message - Failure	Failure to start - $$BlockComment$$
2	PLC Process Control Message - Failure	Failure to stop - $$BlockComment$$
3	PLC Process Control Message - Failure	Uncommanded stop or rejected start command (due to interlock) - $$BlockComment$$

Assignment of auxiliary process values to the block parameters

Value	Block Parameter
@1%s@	BA_NA
@2%d@	BA_ID
@3%d@	STEP_NO

2.12.1.1 MOTOR_A_ALM_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
PROOF	BOOL	Running status feedback signal from the motor	False	True	
A_STRT	BOOL	Automatic Start command input	False	False	
A_STOP	BOOL	Automatic Stop command input	False	False	
PERMIT	BOOL	Permitted to Start input	True	True	
MODE	BOOL	Mode input from of a sequential function chart	False	False	
INT01	BOOL	Interlock 1	True	True	
INT02	BOOL	Interlock 2	True	True	
INT03	BOOL	Interlock 3	True	True	
INT04	BOOL	Interlock 4	True	True	
INT05	BOOL	Interlock 5	True	True	
INT06	BOOL	Interlock 6	True	True	
INT07	BOOL	Interlock 7	True	True	
INT08	BOOL	Interlock 8	True	True	
INT09	BOOL	Interlock 9	True	True	
INT10	BOOL	Interlock 10	True	True	
INT11	BOOL	Interlock 11	True	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
PERMIT_DESC	STRING[32]	Permit Description	''	True	
INT01_DESC	STRING[32]	Interlock 1 Description	''	True	
INT02_DESC	STRING[32]	Interlock 2 Description	''	True	
INT03_DESC	STRING[32]	Interlock 3 Description	''	True	
INT04_DESC	STRING[32]	Interlock 4 Description	''	True	
INT05_DESC	STRING[32]	Interlock 5 Description	''	True	
INT06_DESC	STRING[32]	Interlock 6 Description	''	True	
INT07_DESC	STRING[32]	Interlock 7 Description	''	True	
INT08_DESC	STRING[32]	Interlock 8 Description	''	True	
INT09_DESC	STRING[32]	Interlock 9 Description	''	True	
INT10_DESC	STRING[32]	Interlock 10 Description	''	True	
INT11_DESC	STRING[32]	Interlock 11 Description	''	True	
A1EN	BOOL	Enable Alarm 1	True	True	
A2EN	BOOL	Enable Alarm 2	True	True	
A3EN	BOOL	Enable Alarm 3	True	True	
A4EN	BOOL	Enable Alarm 4	False	True	
OOS	BOOL	Out of Service indicator	False	True	
EV_ID	DWORD	Event ID	Compiler generated for the type or N/A if used as chart *	False	
RSTRT_AUTO_FLAG	BOOL	Default retain on WS, Use load val on CS. Force: Use RSTRT_AUTO_VALUE on restart	False	True	
RSTRT_AUTO_VALUE	BOOL	Init restart value for Auto Mode if RSTRT_AUTO_FLAG true	False	False	
RSTRT_OUT_FLAG	BOOL	Default retain on WS, Use load val on CS. Force: Use RSTRT_AUTO_VALUE on restart	False	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
RSTRT_OUT_VALUE	BOOL	Init restart value for OUT if RSTRT_OUT_FLAG true	False	False	
BA_EN	BOOL	Batch enabled	False	True	
OCCUPIED	BOOL	Batch occupied	False	True	
BA_ID	DWORD	Batch ID number	0	True	
BA_NA	STRING[32]	Batch name	‘‘	True	
STEP_NO	DWORD	Batch step number	0	True	
ConfirmChange	BOOL	If TRUE changes from HMI must be confirmed	False	True	
LIOP_MAN_SEL	BOOL	Select AUTO/MANUAL: 1 = Linking, 0 = Operator active	False	False	
AUT_L	BOOL	Linkable input for AUTO/MANUAL mode, 1 = AUTO	False	False	

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
OUT	BOOL	Output used to ultimately control the motor	False	True	
INTRLK	BOOL	Interlock OK output	False	True	
READY	BOOL	Ready output	False	False	
STATE	INT	State of the motor	0	False	
ALARM_STATE	INT	Alarm state	0	False	
ALM1	BOOL	Alarm 1	False	False	
ALM2	BOOL	Alarm 2	False	False	
ALM3	BOOL	Alarm 3	False	False	
ALM4	BOOL	Alarm 4	False	False	

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
FAIL	BOOL	Indicates whether motor has fault condition	False	True	
AUTO	BOOL	Auto mode output	False	True	
OVERRD	BOOL	Override output	False	True	
START	BOOL	Start command value for manual mode	False	True	
STOP	BOOL	Stop command value for manual mode	False	True	
STRT_T	TIME	Maximum time required to start the motor	T#10S	True	
STOP_T	TIME	Maximum time required to stop the motor	T#10S	True	
AUTO_TRACK	BOOL	If true, block is set to auto mode when SFC transitions to AUTO	False	False	
MODE_HOLD_AUTO	BOOL	Indicates whether block held in auto mode	False	False	
ACCEPT_ANY_AUTO	BOOL	When true, block is set to auto when A_S TRT or A_STOP changes state	False	False	
POS_TRANS_ASTOP	BOOL	Indicates whether a positive transition is required to stop the motor	False	False	
AUTO_TRANS_REQ	BOOL	When true the A_STRT and A_STOP inputs need to change state to issue start/stop	False	False	
FAIL_ON_INTRLK	BOOL	Fail on interlock	True	False	
FAIL_ON_STRT_CMD	BOOL	Fail on start command	True	False	
A1LIM	BOOL	Alarm 1 Limit	True	False	
A1DELAYIN	TIME	Into Alarm 1 Delay Time	T#0MS	False	0 to 24d20h31m23s647ms
A1DELAYOUT	TIME	Out of Alarm 1 Delay Time	T#0MS	False	0 to 24d20h31m23s647ms
A2LIM	BOOL	Alarm 2 Limit	True	False	
A2DELAYIN	TIME	Into Alarm 2 Del ay Time	T#0MS	False	0 to 24d20h31m23s647ms
A2DELAYOUT	TIME	Out of Alarm 2 Delay Time	T#0MS	False	0 to 24d20h31m23s647ms
A3LIM	BOOL	Alarm 3 Limit	True	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
A3DELAYIN	TIME	Into Alarm 3 Delay Time	T#0MS	False	0 to 24d20h31m23s647ms
A3DELAYOUT	TIME	Out of Alarm 3 Delay Time	T#0MS	False	0 to 24d20h31m23s647ms
A4LIM	BOOL	Alarm 4 Limit	True	False	
A4DELAYIN	TIME	Into Alarm 4 Delay Time	T#0MS	False	0 to 24d20h31m23s647ms
A4DELAYOUT	TIME	Out of Alarm 4 Delay Time	T#0MS	False	0 to 24d20h31m23s647ms
DISABLE_ON_OOS	BOOL	Disable on Out of Service	False	False	

2.13 MOTOR_1OUT_AFB

2.13.1 MOTOR_1OUT: Motor with One Output

Function Block Name (compiled version of the Chart)	MOTOR_1OUT_AFB
Function Block Number	FB606
Chart Name	MOTOR_1OUT
See also, Template Chart (usage example)	**MOTOR_1OUT_AT**

Function

The Motor with One Output (MOTOR_1OUT) is intended for use as a start/stop motor with one output. The block is designed for a motor with auto/manual operation, pulse commands, and interlocking. An alarm is provided for failure to reach the desired state.

The [A_STRT] and [A_STOP] inputs are intended to input command pulses from a sequence. The local variable AUTO determines the source of the start/stop commands. In auto mode (AUTO=TRUE) the auto parameters [A_STRT] and [A_STOP] are active, and in manual mode (AUTO=FALSE) the parameters START and STOP are active. START, STOP, and AUTO are to be manipulated via the operator interface. START and STOP are to be set TRUE; they are reset automatically.

A positive transition of auto start [A_STRT] or manual start [START] (depending on the mode) latches the output [OUT]. Interlock [INTRLK] must be TRUE to allow the latch. A positive transition of auto stop [A_STOP] or manual stop STOP (also mode-dependent) unlatches the output. The start and stop timers determine if the motor has failed to reach its desired state, and the failure output [FAIL] will become TRUE accordingly, which triggers the alarm.

Although the MOTOR_1_OUT block was implemented in ladder logic in APACS, it has been implemented in SCL in PCS 7 due to differences between APACS+ and PCS 7 ladder logic. The MOTOR_1OUT is a logical and functional equivalent to the corresponding APACS+ ladder networks. Comments in the provided SCL source code relate elements in the SCL code to corresponding elements in the original APACS+ ladder network.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32). Additionally, it is called in OB100 and in OB102 (see Start-up Characteristics).

Start-up Characteristics

On a warm restart, OB100 is called and parameters retain the values they had immediately before the restart. On a cold restart, OB102 is called and parameters are initialized with the last downloaded values.

The warm and cold startup behavior of AUTO and OUT is configurable, allowing the user to override the default PCS 7 behavior described above if necessary. As delivered, AUTO will be forced to "Manual" and OUT will be forced to "False" on a CPU restart.

To force the AUTO parameter to a specified value on a restart, set the RSTRT_AUTO_FLAG to true ("Force AUTO") and configure RSTRT_AUTO_VALUE with the desired forced value ("Auto" or "Manual"). The RSTRT_AUTO_VALUE will be used to initialize AUTO, if and only if the RSTRT_AUTO_FLAG is true. If RSTRT_AUTO_FLAG is set to false ("Default"), the AUTO parameter will be set in accordance with default PCS 7 behavior.

To force the OUT parameter to a specified value on a restart, set the RSTRT_OUT_FLAG to true ("Force OUT") and configure RSTRT_OUT_VALUE with the desired forced value ("True" or "False"). The RSTRT_OUT_VALUE will be used to initialize OUT, if and only if the RSTRT_OUT_FLAG is true. If RSTRT_OUT_FLAG is set to false ("Default"), the OUT parameter will be set in accordance with default PCS 7 behavior.

Warning

Using "Default" for the RSTRT_OUT_FLAG could result in your output device retaining its last operating state after a restart. Similarly, using "Default" for the RSTRT_AUTO_FLAG could result in your process running in automatic mode after a restart. Use "Forced" for the restart flags and configure the associated restart values to ensure a known restart state.

If restart parameter values are changed offline, the configuration must be compiled (changes only) and downloaded (changes only) to the controller in order for the settings to take effect upon restart. If restart parameter values are changed online (via CFC Test mode), the configuration must be compiled (entire program) and downloaded (entire program) to the controller in order for the settings to take effect upon cold restart. This step is not necessary for the online changes to take effect upon a warm restart.

Operator Control and Monitoring (M&C Relevant)

See Faceplate Information

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

Message #	Message Class	Default Message Text
1	PLC Process Control Message - Failure	Motor Feedback Failure - $$BlockComment$$

Assignment of auxiliary process values to the block parameters

Value	Block Parameter
@1%s@	BA_NA
@2%d@	BA_ID
@3%d@	STEP_NO

2.13.1.1 MOTOR_1OUT_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
PROOF	BOOL	PROOF is used as feedback to confirm that the motor is running	False	True	
A_STRT	BOOL	A_STRT is intended to allow internal logic to automatically start the motor	False	False	
A_STOP	BOOL	A_STOP is intended to allow internal logic to automatically stop the motor	False	False	
INTRLK	BOOL	INTRLK forbids activation of the motor	False	True	
SAMPLE_T	REAL	Sample time (in seconds)	0.1	False	
RSTRT_AUTO_FLAG	BOOL	Default retain on WS, Use load val on CS. Force: Use RSTRT_AUTO_VALUE on restart	False	True	
RSTRT_AUTO_VALUE	BOOL	Init restart value for Auto Mode if RSTRT_AUTO_FLAG true	False	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
RSTRT_OUT_FLAG	BOOL	Default retain on WS, Use load val on CS. Force: Use RSTRT_AUTO_VALUE on restart	False	True	
RSTRT_OUT_VALUE	BOOL	Init restart value for OUT if RSTRT_OUT_FLAG true	False	False	
EV_ID	DWORD	Event ID	Compiler generated	False	
LIOP_MAN_SEL	BOOL	Select AUTO/MANUAL: 1 = Linking, 0 = Operator active	False	False	
AUT_L	BOOL	Linkable input for AUTO/MANUAL mode, 1 = AUTO	False	False	
BA_EN	BOOL	Batch enabled	False	True	
OCCUPIED	BOOL	Batch occupied	False	True	
BA_ID	DWORD	Batch ID number	0	True	
BA_NA	STRING[32]	Batch name	‘‘	True	
STEP_NO	DWORD	Batch step number	0	True	
ConfirmChange	BOOL	If TRUE changes from HMI must be confirmed	False	True	

Note

This list differs from the inputs of the MOTOR_1OUT_AFB block in APACS+ as follows:

- Restart parameters (RSTRT_...) did not exist in APACS+.
- EV_ID and SAMPLE_T did not exist in APACS+.

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
OUT	BOOL	OUT is used to actuate the motor	False	True	
ALARM	BOOL	ALARM is used to signal internal logic when the Motor is in alarm	False	False	
FAIL	BOOL	FAIL motor failed to start or stop after a start or stop request	False	True	
START_FAIL	BOOL	START_FAIL is used to indicate if the motor failed to start	False	False	

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
AUTO	BOOL	AUTO set auto mode (use A_STRT, A_STOP) vs manual mode (use START, STOP)	False	True	
A1EN	BOOL	A1EN is used to enable /disable the internal alarm	True	True	
A1LIM	BOOL	Alarm 1 limit	True	False	
A1DELAYIN	TIME	Into alarm 1 delay time	T#0MS	False	0 to 24d20h31m23s647ms
A1DELAYOUT	TIME	Out of alarm 1 delay time	T#0MS	False	0 to 24d20h31m23s647ms
OOS	BOOL	OOS is used to signal that the internal alarm is Out Of Service	False	True	
DISABLE_ON_OOS	BOOL	Disable on Out of Service	False	False	
START	BOOL	START is used to start the motor when in Manual mode (AUTO=false)	False	True	
STOP	BOOL	STOP is used to stop the motor when in Manual mode (AUTO=false)	False	True	
STRT_T	TIME	START_TIME sets time limit for PROOF to go true	T#10S	True	
STOP_T	TIME	STOP_TIME sets time limit for PROOF to go false	T#10S	True	

2.14 MOTOR_2OUT_AFB

2.14.1 MOTOR_2OUT: Motor with Two Outputs

Function Block Name (compiled version of the Chart)	MOTOR_2OUT_AFB
Function Block Number	FB607
Chart Name	MOTOR_2OUT
See also, Template Chart (usage example)	**MOTOR_2OUT_AT**

Function

The Motor with Two Outputs (MOTOR_2OUT) block is intended for use as a start/stop motor with a start output and a stop output. The block is designed for a motor with auto/manual operation, pulse commands, and interlocking. An alarm is provided for failure to reach the desired state.

The [A_STRT] and [A_STOP] inputs are intended to input command pulses from a sequence. The parameter AUTO determines the source of the start/stop commands. In auto mode (AUTO=TRUE), the auto parameters [A_STRT] and [A_STOP] are active, and in manual mode (AUTO=FALSE) the parameters START and STOP are active. START, STOP, and AUTO are to be manipulated via the operator interface. START and STOP are to be set TRUE; they are reset automatically.

A positive transition of auto start [A_STRT] or manual start [START] (depending on the mode) latches the internal START_LATCH and pulses the start output [STRT_O]. Interlock [INTRLK] must be TRUE to allow the latch. A positive transition of auto stop [A_STOP] or manual stop STOP (also mode-dependent) unlatches the internal START_LATCH and pulses the stop output [STOP_O]. The start and stop timers pulse the outputs and determine if the motor has failed to reach its desired state, and the failure output [FAIL] will become TRUE accordingly, which triggers the alarm.

Although the MOTOR_2OUT block was implemented in ladder logic in APACS, it has been implemented in SCL in PCS 7 due to differences between APACS+ and PCS 7 ladder logic. The MOTOR_2OUT is a logical and functional equivalent to the corresponding APACS+ ladder network. Comments in the provided SCL source code relate elements in the SCL code to corresponding elements in the original APACS+ ladder network.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32). Additionally, it is called in OB100 and in OB102 (see Start-up Characteristics).

Start-up Characteristics

On a warm restart, OB100 is called and parameters retain the values they had immediately before the restart. On a cold restart, OB102 is called and parameters are initialized with the last downloaded values.

The warm and cold startup behavior of AUTO is configurable, allowing the user to override the default PCS 7 behavior described above if necessary. As delivered, AUTO will be forced to "Manual" and OUT will be forced to "False" on a CPU restart.

To force the AUTO parameter to a specified value on a restart, set the RSTRT_AUTO_FLAG to true ("Force AUTO") and configure RSTRT_AUTO_VALUE with the desired forced value ("Auto" or "Manual"). The RSTRT_AUTO_VALUE will be used to initialize AUTO, if and only if the RSTRT_AUTO_FLAG is true. If RSTRT_AUTO_FLAG is set to false ("Default"), the AUTO parameter will be set in accordance with default PCS 7 behavior.

Warning

Using "Default" for the RSTRT_AUTO_FLAG could result in your process running in automatic mode after a restart. Use "Forced" for the restart flag and configure the associate restart value to ensure a known restart state.

The STRT_O and STOP_O outputs will be automatically cleared on a restart.

If restart parameter values are changed offline, the configuration must be compiled (changes only) and downloaded (changes only) to the controller in order for the settings to take effect upon restart. If restart parameter values are changed online (via CFC Test mode), the configuration must be compiled (entire program) and downloaded (entire program) to the controller in order for the settings to take effect upon cold restart. This step is not necessary for the online changes to take effect upon a warm restart.

Operator Control and Monitoring (M&C Relevant)

See Faceplate Information

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

Message #	Message Class	Default Message Text
1	PLC Process Control Message - Failure	Motor Feedback Failure - $$BlockComment$$

Assignment of auxiliary process values to the block parameters

Value	Block Parameter
@1%s@	BA_NA
@2%d@	BA_ID
@3%d@	STEP_NO

2.14.1.1 MOTOR_2OUT_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
PROOF	BOOL	PROOF is used as feedback to confirm that the motor is running	False	True	
A_STRT	BOOL	A_STRT is intended to allow internal logic to automatically start the motor	False	False	
A_STOP	BOOL	A_STOP is intended to allow internal logic to automatically stop the motor	False	False	
INTRLK	BOOL	INTRLK forbids activation of the motor	False	True	
SAMPLE_T	REAL	Sample time (in seconds)	Compiler generated	False	
RSTRT_AUTO_FLAG	BOOL	Default retain on WS, Use load val on CS. Force: Use RSTRT_AUTO_VALUE on restart	False	True	
RSTRT_AUTO_VALUE	BOOL	Init restart value for Auto Mode if RSTRT_AUTO_FLAG true	False	False	
EV_ID	DWORD	Event ID	Compiler generated	False	
LIOP_MAN_SEL	BOOL	Select AUTO/MANUAL: 1 = Linking, 0 = Operator active	False	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
AUT_L	BOOL	Linkable input for AUTO/MANUAL mode, 1 = AUTO	False	False	
BA_EN	BOOL	Batch enabled	False	True	
OCCUPIED	BOOL	Batch occupied	False	True	
BA_ID	DWORD	Batch ID number	0	True	
BA_NA	STRING[32]	Batch name	‘‘	True	
STEP_NO	DWORD	Batch step number	0	True	
ConfirmChange	BOOL	If TRUE changes from HMI must be confirmed	False	True	

Note

This list differs from the inputs of the MOTOR_2OUT_AFB block in APACS+ as follows:

- Restart parameters (RSTRT_...) did not exist in APACS+.
- EV_ID and SAMPLE_T did not exist in APACS+.

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
STRT_O	BOOL	STRT_O is used to actuate the motor start	False	False	
STOP_O	BOOL	STOP_O is used to actuate the motor stop	False	False	
ALARM	BOOL	ALARM is used to signal internal logic when the Motor is in alarm	False	False	
FAIL	BOOL	FAIL motor failed to start or stop after a start or stop request	False	True	
OUT	BOOL	OUT is used to let the OS 'see' the internal START_LATCH coil	False	True	
START_FAIL	BOOL	START_FAIL is used to indicate if the motor failed to start	False	False	

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
AUTO	BOOL	AUTO set auto mode (use A_STRT, A_STOP) vs manual mode (use START, STOP)	False	True	
A1EN	BOOL	A1EN is used to enable /disable the internal alarm	True	True	
A1LIM	BOOL		True	True	
A1DELAYIN	TIME	Into alarm 1 delay time	T#0MS	False	0 to 24d20h31m 23s647ms
A1DELAYOUT	TIME	Out of alarm 1 delay time	T#0MS	False	0 to 24d20h31m 23s647ms
OOS	BOOL	OOS is used to signal that the internal alarm is Out Of Service	True	False	
DISABLE_ON_OOS	BOOL	Disable on Out of Service	False	False	
START	BOOL	START is used to start the motor when in Manual mode (AUTO=false)	True	False	
STOP	BOOL	STOP is used to stop the motor when in Manual mode (AUTO=false)	True	False	
STRT_T	TIME	START_TIME sets time limit for PROOF to go true	T#10S	True	
STOP_T	TIME	STOP_TIME sets time limit for PROOF to go false	T#10S	True	

2.15 PID_AFB

2.15.1 PID_AFB: Proportional-Integral-Derivative

Function Block Name	PID_AFB
Function Block Number	FB501

Function

The PID_AFB performs Proportional-Integral-Derivative (PID) control functions. It can also be used as the basis for a PI controller by setting the derivative time value to 0.0 min. The block utilizes reset feedback, allowing for various control schemes including override, reset limiting, etc.

When the AUTO value = FALSE, the block operates in tracking mode (OUT value = FDBK value). The PID block inputs (PV, SP & FDBK) and its outputs (OUT, ERR) are in engineering units. The PID equation is defined using normalized values. Engineering values are converted to and from normalized values with user-defined operating limits for both the process input and process output. The operating limits are obtained from the process flowsheets or sensor/actuator ranges. Any transition of the INIT input causes execution of the initialization equations which recalculate the reset component so that the OUT value does not experience a bump due to proportional action.

Note

Normal limits for the TI In/Out value are from 0.001 to 4000.0. In the special case of TI = 4000.0, all integral action is removed from the output calculation.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32).

Start-up Characteristics

This function block has no special startup properties.

Operator Control and Monitoring (M&C Relevant)

There is no associated faceplate for this block.

2.15.1.1 PID_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
PV	REAL	Process variable input	0	True	
SP	REAL	Setpoint input	0	False	
FDBK	REAL	Reset feedback input	0	False	
AUTO	BOOL	Track command input. When Manual, the block is in a tracking mode [OUT=FDBK].	False	False	
INIT	BOOL	Transitions of the INIT input cause the initialization equation to be performed.	False	False	
SAMPLE_T	REAL	Current scan rate supplied by compiler		False	

Note

This list differs from the inputs of the PID_AFB block in APACS+ as follows:

- The SAMPLE_T input is required for the block to obtain the scan rate.

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
OUT	REAL	Calculated output. It is limited from -3.3 to +103.3% of range EOUTL to EOUTH.	0	False	
ERR	REAL	Error output [(PV) – (SP)].		False	

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
PG	REAL	Proportional Gain	0.001	True	0.001 to 100.0
TI	REAL	Time-Integral Minutes per Reset	4000.0	True	0.0 to 4000.0
TD	REAL	Time-Derivative Minutes	0	True	0.00 to 100.0
DG	REAL	Derivative Gain	10.0	True	2.00 to 20.00
DIR_ACT	BOOL	Reverse or Direct action	False	True	
EINL	REAL	Engineering unit: Input, Low	0	True	
EINH	REAL	Engineering unit: Input, High	100.0	True	
EOUTL	REAL	Engineering unit: Output, Low	0	True	
EOUTH	REAL	Engineering unit: Output, High	100.0	True	
ADAPTIVE_PG_INT	BOOL	Adaptive Gain Init flag	True	False	

2.16 PRIMARY_AFB

2.16.1 PRIMARY: Primary Controller

Function Block Name (compiled version of the Chart)	PRIMARY_AFB
Function Block Number	FB608
Chart Name	PRIMARY
See also, Template Chart (usage example)	**PRIMARY_SECONDARY_AT**

Function

The Primary Controller (PRIMARY_AFB) block is a factory-compiled function block. It is intended for use as a primary single-loop PID controller with tracking setpoint as part of a cascade control circuit. The PRIMARY_AFB block is designed to be used with the SECONDARY_AFB block to form a cascade circuit when it is preferable to control the primary and secondary loops separately. Normally, the output from the primary loop is to be the setpoint to the secondary loop, and the cascade status of the secondary loop is to be fed to the primary loop. High and low alarms are provided for the process variable, and an alarm is provided for absolute deviation between process variable and setpoint.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32). Additionally, it is called in OB100 and in OB102 (see Start-up characteristics).

Start-up Characteristics

On a warm restart, OB100 is called and parameters retain the values they had immediately before the restart. On a cold restart, OB102 is called and parameters are initialized with the last downloaded values.

The warm and cold startup behavior of AUTO, SP, and OUT is configurable, allowing the user to override the default PCS 7 behavior described above, if necessary. As delivered, AUTO will be forced to "Manual" and OUT will be forced to 0.0 on a CPU restart. SP will be set in accordance with default PCS 7 behavior.

To force the AUTO parameter to a specified value on a restart, set the RSTRT_AUTO_FLAG to true ("Force AUTO") and configure RSTRT_AUTO_VALUE with the desired forced value ("Auto" or "Manual"). The RSTRT_AUTO_VALUE will be used to initialize AUTO, if and only if the RSTRT_AUTO_FLAG is true. If RSTRT_AUTO_FLAG is set to false ("Default"), the AUTO parameter will be set in accordance with default PCS 7 behavior.

To force the SP parameter to a specified value on a restart, set the RSTRT_SP_FLAG to true ("Force SP") and configure RSTRT_SP_VALUE with the

desired forced value. The RSTRT_SP_VALUE will be used to initialize SP, if the RSTRT_SP_FLAG is true and either

- The block is forced to start in Auto mode, or
- AUTO is not forced and the block restarts with Auto mode (AUTO="Auto") and in secondary cascade (CASC_S = "True").

If RSTRT_SP_FLAG is set to false ("Default"), the SP parameter will be set in accordance with default PCS 7 behavior.

To force the OUT parameter to a specified value on a restart, set the RSTRT_OUT_FLAG to true ("Force OUT") and configure RSTRT_OUT_VALUE with the desired forced value. The RSTRT_OUT_VALUE will be used to initialize OUT, if and only if the RSTRT_OUT_FLAG is true and the block restarts in Manual mode (AUTO = "Manual"). If RSTRT_OUT_FLAG is set to false ("Default"), the OUT parameter will be set in accordance with default PCS 7 behavior.

Warning

Using "Default" for the RSTRT_OUT_FLAG could result in your output device retaining its last operating state after a restart. Similarly, using "Default" for the RSTRT_AUTO_FLAG could result in your process running in automatic mode after a restart. Use "Forced" for the restart flags and configure the associated restart values to ensure a known restart state.

If restart parameter values are changed offline, the configuration must be compiled (changes only) and downloaded (changes only) to the controller in order for the settings to take effect upon restart. If restart parameter values are changed online (via CFC Test mode), the configuration must be compiled (entire program) and downloaded (entire program) to the controller in order for the settings to take effect upon cold restart. This step is not necessary for the online changes to take effect upon a warm restart.

Operator Control and Monitoring (M&C Relevant)

See Faceplate Information

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

ALARM_8P ID	Message #	Message Class	Default Message Text
EV_ID	1	Warning - above	HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID	2	Warning - below	LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID	3	Alarm - above	HIGH_DEV - $$BlockComment$$ - Limit = @6%f@
EV_ID	4	Alarm - below	LOW_DEV - $$BlockComment$$ - Limit = @7%f@
EV_ID	5	Alarm - above	ABSOLUTE_DEV - $$BlockComment$$ - Limit = @8%f@
EV_ID	6	PLC Process Control Message - Failure	OUT_OF_RANGE - $$BlockComment$$
EV_ID	7	PLC Process Control Message - Failure	QUALITY_NOT_GOOD - $$BlockComment$$
EV_ID2	1	Alarm - above	HIGH_ HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID2	2	Alarm - below	LOW_ LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID2	3	Warning - above	INC_ROC - $$BlockComment$$ - Limit = @6%f@
EV_ID2	4	Warning - below	DEC_ROC - $$BlockComment$$ - Limit = @7%f@
EV_ID2	5	Warning - above	ABS_ROC - $$BlockComment$$ - Limit = @8%f@

Assignment of auxiliary process values to the block parameters

ALARM_8P ID	Value	Block Parameter
EV_ID	@1%s@	BA_NA
EV_ID	@2%d@	BA_ID
EV_ID	@3%d@	STEP_NO
EV_ID	@4%f@	H_LIM
EV_ID	@5%f@	L_LIM
EV_ID	@6%f@	HD_LIM
EV_ID	@7%f@	LD_LIM
EV_ID	@8%f@	AD_LIM
EV_ID2	@1%s@	BA_NA
EV_ID2	@2%d@	BA_ID
EV_ID2	@3%d@	STEP_NO
EV_ID2	@4%f@	HH_LIM
EV_ID2	@5%f@	LL_LIM
EV_ID2	@6%f@	IR_LIM
EV_ID2	@7%f@	DR_LIM
EV_ID2	@8%f@	AR_LIM

2.16.1.1 PRIMARY_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
PV	REAL	Process Variable	0	True	
FDBK	REAL	Controller tracks feedback when (AUTO=FALSE) or (CASC_S=FALSE)	0	False	
CASC_S	BOOL	Secondary cascade status	False	False	
INIT	BOOL	Transitions of the INIT input cause the initialization equation to be performed	False	False	
TSP	REAL	SP target value for SP ramping	0	True	
H_EN	BOOL	Enable High Alarm	True	True	
L_EN	BOOL	Enable Low Alarm	True	True	
HD_EN	BOOL	Enable High Deviation Alarm	False	True	
LD_EN	BOOL	Enable Low Deviation Alarm	False	True	
AD_EN	BOOL	Enable Absolute Deviation Alarm	True	True	
OOR_EN	BOOL	Enable Out of Range Alarm	False	True	
QUAL_EN	BOOL	Enable Quality Not Good Alarm	False	True	
HH_EN	BOOL	Enable High High Alarm	False	True	
LL_EN	BOOL	Enable Low Low Alarm	False	True	
IR_EN	BOOL	Enable Increasing Rate of Change Alarm	False	True	
DR_EN	BOOL	Enable Decreasing Rate of Change Alarm	False	True	
AR_EN	BOOL	Enable Absolute Rate of Change Alarm	False	True	
OOS	BOOL	Out of Service indicator	False	True	
QBAD	BOOL	Bad quality indicator	False	False	
EV_ID	DWORD	Event ID	DW#16#5	False	
EV_ID2	DWORD	Event ID	DW#16#6	False	
SAMPLE_T	REAL	Scan Rate. Use only when compiling CFC as block type.	0.1	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
RSTRT_AUTO_FLAG	BOOL	Default: Retain on WS, Use load value on CS. Force: Use RSTRT_AUTO_VALUE on restart	True	False	
RSTRT_AUTO_VALUE	BOOL	Init restart value for Auto Mode if RSTRT_AUTO_FLAG true	False	False	
RSTRT_SP_FLAG	BOOL	Default: Retain on WS, Use load value on CS. Force: Use RSTRT_SP_VALUE restart.	False	False	
RSTRT_SP_VALUE	REAL	Init restart value for SP if RSTRT_SP_FLAG true and Auto mode.	0	False	
RSTRT_OUT_FLAG	BOOL	Default: Retain on WS, Use load value on CS. Force: Use RSTRT_OUT_VALUE on restart.	True	False	
RSTRT_OUT_VALUE	REAL	Init restart value for OUT if RSTRT_OUT_FLAG true and manual mode	0	False	
BA_EN	BOOL	Batch enabled	False	True	
OCCUPIED	BOOL	Batch occupied	False	True	
BA_ID	DWORD	Batch ID number	0	True	
BA_NA	STRING[32]	Batch name	‘‘	True	
STEP_NO	DWORD	Batch step number	0	True	
DecimalPlacesPV	INT	Decimal places displayed in HMI for PV, SP etc.	2	True	
DecimalPlacesOUT	INT	Decimal places displayed in HMI for OUT, OUT Limits	2	True	
ConfirmChange	BOOL	If TRUE changes from HMI must be confirmed	False	True	
LIOP_MAN_SEL	BOOL	Select AUTO/MANUAL: 1 = Linking, 0 = Operator active	0	False	
AUT_L	BOOL	Linkable input for AUTO/MANUAL mode, 1 = AUTO	0	False	
LIOP_ ISP_SEL	BOOL	Select internal SP: 1 = from Link, 0 = from Operator	0	False	
SP_L	REAL	Linkable input for internal Setpoint	0.0	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
LIOP_OUT_SEL	BOOL	Select writing to OUT: 1 = from Link, 0 = from Operator	0	False	
OUT_L	REAL	Linkable input to be written to OUT	0.0	False	

Note

This list differs significantly from the inputs of the PRIMARY_AFB block in APACS+. In APACS+, only the PV, FDBK and CASC_S inputs are present.

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
SP_OUT	REAL	Calculated set point value	0	False	
OUT	REAL	Calculated output variable that has been limited	0	False	
ALARM	BOOL	Alarm Status	False	False	
SS	BOOL	Secondary Standby flag	True	True	
H_ALM	BOOL	When TRUE, the High alarm is detected and enabled.	False	False	
L_ALM	BOOL	When TRUE, the Low alarm is detected and enabled.	False	False	
HD_ALM	BOOL	When TRUE, the High Deviation alarm is detected and enabled.	False	False	
LD_ALM	BOOL	When TRUE, the Low Deviation alarm is detected and enabled.	False	False	
AD_ALM	BOOL	When TRUE, the Absolute Deviation alarm is detected and enabled.	False	False	
OOR_ALM	BOOL	When TRUE, the Out of Range alarm is detected and enabled.	False	False	
QUAL_ALM	BOOL	When TRUE, the Quality Not Good alarm is detected and enabled.	False	False	
HH_ALM	BOOL	When TRUE, the High High alarm is detected and enabled.	False	False	
LL_ALM	BOOL	When TRUE, the Low Low alarm is detected and enabled.	False	False	
IR_ALM	BOOL	When TRUE, the Increasing Rate of Change alarm is detected and enabled.	False	False	
DR_ALM	BOOL	When TRUE, the Decreasing Rate of Change alarm is detected and enabled.	False	False	
AR_ALM	BOOL	When TRUE, the Absolute Rate of Change alarm is detected and enabled.	False	False	

Note

The APACS+ SP output has been converted to SP_OUT. SP_OUT is used to interconnect the set-point value to other entities on a chart. The OS interaction is handled by the SP InOut parameter.

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
AUTO	BOOL	Auto/manual mode control	False	True	
SP	REAL	Computer set point value	0	True	
RAMP	BOOL	SP ramping on/off mode control	False	True	
R_TIME	TIME	Ramp time	T#0MS	True	0 to 24d20h31m 23s647ms
SPL	REAL	Set point low limit	0	True	
SPH	REAL	Set point high limit	100	True	
R_RATE	REAL	Ramp rate (in engineering units per min)	0	True	
USE_RR	BOOL	Ramp mode control. Use RTIME or RRATE to control ramp	False	True	
PV_TRACK	BOOL	Internal setpoint tracking control. When Track, internal setpoint tracks PV	True	True	
PG	REAL	Proportional Gain	0.001	True	0.001 to 100.0
TI	REAL	Time-Integral Minutes per Reset	4000	True	0.0 to 4000.0
TD	REAL	Time-Derivative Minutes	0	True	0.00 to 100.0
DG	REAL	Derivative Gain	10	True	2.00 to 20.00
DIR	BOOL	Reverse or Direct action	False	True	
ADAPTIVE_PG_INIT	BOOL	Adaptive Gain Init flag	True	False	
PV_EUMIN	REAL	Engineering unit: Input, Low	0	True	
PV_EUMAX	REAL	Engineering unit: Input, High	100	True	
VL_EUMIN	REAL	Engineering unit: Output, Low	0	True	
VL_EUMAX	REAL	Engineering unit: Output, High	100	True	
VL	REAL	Computer Output, typically written to by an operator command	0	True	
VLL	REAL	OUT low limit input	-3.3	True	
VLH	REAL	OUT high limit input	103.3	True	
H_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the High alarm.	100	True	
H_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the High alarm to clear.	0.5	True	
H_DELAYIN	TIME	The amount of time the input must reach or exceed the High limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms

Name	Data Type	Explanation	Initial	OC&M	Valid Values
H_DELAYOUT	TIME	The amount of time the input must be within the High limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
L_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the Low alarm.	0	True	
L_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Low alarm to clear.	0.5	True	
L_DELAYIN	TIME	The amount of time the input must reach or exceed the Low limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
L_DELAYOUT	TIME	The amount of time the input must be within the Low limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
HD_LIM	REAL	The value (in engineering units) to which (PV) – (DEV) is compared for the High Deviation alarm.	0	True	
HD_DB	REAL	The amount of engineering units by which (PV) – (DEV) must be within the limit for the High Deviation alarm to clear.	0.5	True	
HD_DELAYIN	TIME	The amount of time the input must reach or exceed the High Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
HD_DELAYOUT	TIME	The amount of time the input must be within the High Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
LD_LIM	REAL	The value (in engineering units) to which (DEV) - (PV) is compared for the Low Deviation alarm.	0	True	
LD_DB	REAL	The amount of engineering units by which (DEV) - (PV) must be within the limit for the Low Deviation alarm to clear.	0.5	True	
LD_DELAYIN	TIME	The amount of time the input must reach or exceed the Low Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
LD_DELAYOUT	TIME	The amount of time the input must be within the Low Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
AD_LIM	REAL	The value (in engineering units) to which ABS((PV) – (DEV)) is compared for the Absolute Deviation alarm.	100	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
AD_DB	REAL	The amount of engineering units by which ABS((PV) – (DEV)) must be within the limit for the Absolute Deviation alarm to clear.	0.5	True	
AD_DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
AD_DELAYOUT	TIME	The amount of time the input must be within the Absolute Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
HH_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the High High alarm.	0	True	
HH_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the High High alarm to clear.	0.5	True	
HH_DELAYIN	TIME	The amount of time the input must reach or exceed the High High limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
HH_DELAYOUT	TIME	The amount of time the input must be within the High High limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
LL_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the Low Low alarm.	0	True	
LL_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Low Low alarm to clear.	0.5	True	
LL_DELAYIN	TIME	The amount of time the input must reach or exceed the Low Low limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
LL_DELAYOUT	TIME	The amount of time the input must be within the Low Low limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
IR_LIM	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Increasing Rate of Change alarm.	0	True	
IR_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Increasing Rate of Change alarm to clear.	0.5	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
IR_DELAYIN	TIME	The amount of time the input must reach or exceed the Increasing Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
IR_DELAYOUT	TIME	The amount of time the input must be within the Increasing Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
DR_LIM	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Decreasing Rate of Change alarm.	0	True	
DR_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Decreasing Rate of Change alarm to clear.	0.5	True	
DR_DELAYIN	TIME	The amount of time the input must reach or exceed the Decreasing Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
DR_DELAYOUT	TIME	The amount of time the input must be within the Decreasing Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
AR_LIM	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Absolute Rate of Change alarm.	0	True	
AR_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Absolute Rate of Change alarm to clear.	0.5	True	
AR_DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
AR_DELAYOUT	TIME	The amount of time the input must be within the Absolute Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
OOR_DELAYIN	TIME	The amount of time the input must be outside the limits EINH, EINL to trip the Out of Range alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
OOR_DELAYOUT	TIME	The amount of time the input must be within the limits EINH, EINL to cause the Out of Range alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms

Name	Data Type	Explanation	Initial	OC&M	Valid Values
QUAL_DELAYIN	TIME	The amount of time the input quality must be bad to trip the Quality Not Good alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
QUAL_DELAYOUT	TIME	The amount of time the input quality must be good to cause the Quality Not Good alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
DISABLE_ON_OOS	BOOL	When set to TRUE, the block is disabled if (OOS) is also TRUE.	False	False	

2.17 RATIO_SET_AFB

2.17.1 RATIO_SET: Ratio Setpoint Controller

Function Block Name (compiled version of the Chart)	RATIO_SET_AFB
Function Block Number	FB609
Chart Name	RATIO_SET
See also, Template Chart (usage example)	**RATIO_SET_AT**

Function

The Ratio Setpoint Controller (RATIO_SET_AFB) block is a factory-compiled function block. It is intended for use as a ratio-set PID controller with tracking setpoint. An external input is multiplied by a desired ratio value (i.e. a controlled variable divided by wild variable) to calculate the external setpoint. The process variable is divided by the external input for display of the actual ratio. Both high and low alarms are provided for the process variable, and an additional alarm is provided to detect an absolute deviation between the process variable and setpoint.

RATIO_SET_AFB is a compiled version of the RATIO_SET source chart. Within the chart, there is a control loop that contains PID_AFB and AM_AFB blocks along with alarming logic. The loop also uses two SETPNT_AFB blocks; the RATIO block provides an input of the ratio calculation, the result of which is the external setpoint of the SETPOINT block.

The RATIO block holds the desired ratio, modified via the computer setpoint (RAT_DES), and the low and high range values of the ratio. The RATIO_MIN and RATIO_MAX values are used by the ACT_RAT_LIMIT block (described below). The ratio input [RAT_INP] is multiplied by the desired ratio (from the RATIO block) to calculate the external setpoint. The SETPOINT block supplies the setpoint [SP] to the CONTROLLER block. The SETPOINT block has a ramping feature for internal mode (EXT=FALSE). In external mode (EXT=TRUE), the calculated external setpoint becomes the controller setpoint, and the SETPOINT block internal setpoint (ISP) tracks the external setpoint, thereby ensuring no change in setpoint during transition to internal mode. The setpoint tracks the process variable [PV] in manual mode (AUTO=FALSE).

The CONTROLLER block is a PID_AFB block. It operates in auto mode (AUTO=TRUE) and tracks the downstream output [OUT] in manual mode (AUTO=FALSE). The AUTO_MANUAL block provides manual control of the output. In auto mode, the CONTROLLER block output becomes the loop output [OUT]. In manual mode, the AUTO_MANUAL block's internal output (IOUT) becomes the loop output.

The ACT_RAT_LIMIT block is used to apply limits to the actual ratio [RAT_ACT], which is the quotient of the process variable and the external input.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32). Additionally, it is called in OB100 and in OB102 (see Start-up Characteristics).

Start-up Characteristics

On a warm restart, OB100 is called and parameters retain the values they had immediately before the restart. On a cold restart, OB102 is called and parameters are initialized with the last downloaded values.

The warm and cold startup behavior of AUTO, SP, RAT_DES, EXT, and OUT is configurable, allowing the user to override the default PCS 7 behavior described above if necessary. As delivered, AUTO will be forced to "Manual", EXT will be forced to "Internal" and OUT wil be forced to 0.0 on a CPU restart. SP and RAT_DES will be set in accordance with default PCS 7 behavior: the value will be retained on a warm restart or set to the last downloaded value on a cold restart.

To force the AUTO parameter to a specified value on a restart, set the RSTRT_AUTO_FLAG to true ("Force AUTO") and configure RSTRT_AUTO_VALUE with the desired forced value ("Auto" or "Manual"). The RSTRT_AUTO_VALUE will be used to initialize AUTO, if and only if the RSTRT_AUTO_FLAG is true. If RSTRT_AUTO_FLAG is set to false ("Default"), the AUTO parameter will be set in accordance with default PCS 7 behavior.

To force the SP parameter to a specified value on a restart, set the RSTRT_SP_FLAG to true ("Force SP") and configure RSTRT_SP_VALUE with the desired forced value. The RSTRT_SP_VALUE will be used to initialize SP, if and only if the RSTRT_SP_FLAG is true and the block restarts in Auto mode (AUTO="Auto"). If RSTRT_SP_FLAG is set to false ("Default"), the SP parameter will be set in accordance with default PCS 7 behavior.

To force the RAT_DES parameter to a specified value on a restart, set the RSTRT_ RAT_DES _FLAG to true ("Force RAT_DES") and configure RSTRT_ RAT_DES _VALUE with the desired forced value. The RSTRT_ RAT_DES _VALUE will be used to initialize RAT_DES, if and only if the RSTRT_ RAT_DES _FLAG is true and the block restarts in Auto mode (AUTO="Auto"). If RSTRT_ RAT_DES _FLAG is set to false ("Default"), the RAT_DES parameter will be set in accordance with default PCS 7 behavior.

To force the EXT parameter to a specified value on a restart, set the RSTRT_EXT_FLAG to true ("Force EXT") and configure RSTRT_EXT_VALUE with the desired forced value ("Internal or External"). The RSTRT_EXT_VALUE will be used to initialize EXT, if and only if the RSTRT_EXT_FLAG is true. If RSTRT_EXT_FLAG is set to false ("Default"), the EXT parameter will be set in accordance with default PCS 7 behavior. Forcing the EXT parameter to "External" will cause the SP to be set to the ESP, thereby overriding the SP restart parameter settings.

To force the OUT parameter to a specified value on a restart, set the RSTRT_OUT_FLAG to true ("Force OUT") and configure RSTRT_OUT_VALUE with the desired forced value. The RSTRT_OUT_VALUE will be used to initialize OUT, if and only if the RSTRT_OUT_FLAG is true and the block restarts in Manual mode (AUTO = "Manual"). If RSTRT_OUT_FLAG is set to false ("Default"), the OUT parameter will be set in accordance with default PCS 7 behavior.

Warning

Using "Default" for the RSTRT_OUT_FLAG could result in your output device retaining its last operating state after a restart. Similarly, using "Default" for the RSTRT_AUTO_FLAG could result in your process running in automatic mode after a restart. Use "Forced" for the restart flags and configure the associated restart values to ensure a known restart state.

If restart parameter values are changed offline, the configuration must be compiled (changes only) and downloaded (changes only) to the controller in order for the settings to take effect upon restart. If restart parameter values are changed online (via CFC Test mode), the configuration must be compiled (entire program) and downloaded (entire program) to the controller in order for the settings to take effect upon cold restart. This step is not necessary for the online changes to take effect upon a warm restart.

Operator Control and Monitoring (M&C Relevant)

See Faceplate Information

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

ALARM_8P ID	Message #	Message Class	Default Message Text
EV_ID	1	Warning - above	HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID	2	Warning - below	LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID	3	Alarm - above	HIGH_DEV - $$BlockComment$$ - Limit = @6%f@
EV_ID	4	Alarm - below	LOW_DEV - $$BlockComment$$ - Limit = @7%f@
EV_ID	5	Alarm - above	ABSOLUTE_DEV - $$BlockComment$$ - Limit = @8%f@
EV_ID	6	PLC Process Control Message - Failure	OUT_OF_RANGE - $$BlockComment$$
EV_ID	7	PLC Process Control Message - Failure	QUALITY_NOT_GOOD - $$BlockComment$$
EV_ID2	1	Alarm - above	HIGH_ HIGH - $$BlockComment$$ - Limit = @4%f@

ALARM_8P ID	Message #	Message Class	Default Message Text
EV_ID2	2	Alarm - below	LOW_ LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID2	3	Warning - above	INC_ROC - $$BlockComment$$ - Limit = @6%f@
EV_ID2	4	Warning - below	DEC_ROC - $$BlockComment$$ - Limit = @7%f@
EV_ID2	5	Warning - above	ABS_ROC - $$BlockComment$$ - Limit = @8%f@

Assignment of auxiliary process values to the block parameters

ALARM_8P ID	Value	Block Parameter
EV_ID	@1%s@	BA_NA
EV_ID	@2%d@	BA_ID
EV_ID	@3%d@	STEP_NO
EV_ID	@4%f@	H_LIM
EV_ID	@5%f@	L_LIM
EV_ID	@6%f@	HD_LIM
EV_ID	@7%f@	LD_LIM
EV_ID	@8%f@	AD_LIM
EV_ID2	@1%s@	BA_NA
EV_ID2	@2%d@	BA_ID
EV_ID2	@3%d@	STEP_NO
EV_ID2	@4%f@	HH_LIM
EV_ID2	@5%f@	LL_LIM
EV_ID2	@6%f@	IR_LIM
EV_ID2	@7%f@	DR_LIM
EV_ID2	@8%f@	AR_LIM

2.17.1.1 RATIO_SET_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
PV	REAL	Process Variable	0	True	
RAT_INP	REAL	External Input. Ratio thereof sets loop's setpoint	0	True	
TSP	REAL	SP target value for SP ramping	0	True	
H_EN	BOOL	Enable High Alarm	True	True	
L_EN	BOOL	Enable Low Alarm	True	True	
HD_EN	BOOL	Enable High Deviation Alarm	False	True	
LD_EN	BOOL	Enable Low Deviation Alarm	False	True	
AD_EN	BOOL	Enable Absolute Deviation Alarm	True	True	
OOR_EN	BOOL	Enable Out of Range Alarm	False	True	
QUAL_EN	BOOL	Enable Quality Not Good Alarm	False	True	
HH_EN	BOOL	Enable High High Alarm	False	True	
LL_EN	BOOL	Enable Low Low Alarm	False	True	
IR_EN	BOOL	Enable Increasing Rate of Change Alarm	False	True	
DR_EN	BOOL	Enable Decreasing Rate of Change Alarm	False	True	
AR_EN	BOOL	Enable Absolute Rate of Change Alarm	False	True	
OOS	BOOL	Out of Service indicator	False	True	
QBAD	BOOL	Bad quality indicator	False	False	
EV_ID	DWORD	Event ID	DW#16#5	False	
EV_ID2	DWORD	Event ID	DW#16#6	False	
SAMPLE_T	REAL	Current scan rate supplied by OS	0.1	False	
RSTRT_AUTO_FLAG	BOOL	Default: Retain on WS, Use load value on CS. Force: Use RSTRT_AUTO_VALUE on restart	True	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
RSTRT_AUTO_VALUE	BOOL	Init restart value for Auto Mode if RSTRT_AUTO_FLAG true	False	False	
RSTRT_EXT_FLAG	BOOL	Default: Retain on WS, Use load value on CS. Force: Use RSTRT_EXT_VALUE on restart	True	False	
RSTRT_EXT_VALUE	BOOL	Init restart value for EXTernal if RSTRT_EXT_FLAG true	False	False	
RSTRT_SP_FLAG	BOOL	Default: Retain on WS, Use load value on CS. Force: Use RSTRT_SP_VALUE on restart.	False	False	
RSTRT_SP_VALUE	REAL	Init restart value for SP if RSTRT_SP_FLAG true and Auto mode.	0	False	
RSTRT_RAT_DES_FLAG	BOOL	Default: Retain on WS, Use load value on CS. Force: Use RSTRT_OUT_VALUE on restart.	False	False	
RSTRT_RAT_DES_VALUE	REAL	Init restart value for RAT_DES if RSTRT_ RAT_DES _FLAG true.	0	False	
RSTRT_OUT_FLAG	BOOL	Default: Retain on WS, Use load value on CS. Force: Use RSTRT_OUT_VALUE on restart.	True	False	
RSTRT_OUT_VALUE	REAL	Init restart value for OUT if RSTRT_OUT_FLAG true and manual mode.	0	False	
BA_EN	BOOL	Batch enabled	False	True	
OCCUPIED	BOOL	Batch occupied	False	True	
BA_ID	DWORD	Batch ID number	0	True	
BA_NA	STRING[32]	Batch name	‘‘	True	
STEP_NO	DWORD	Batch step number	0	True	
DecimalPlacesPV	INT	Decimal places displayed in HMI for PV, SP etc.	2	True	
DecimalPlacesOUT	INT	Decimal places displayed in HMI for OUT, OUT Limits	2	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
DecimalPlacesRATIO	INT	Decimal places displayed in HMI for RAT_ACT, RAT_DES	2	True	
ConfirmChange	BOOL	If TRUE changes from HMI must be confirmed	False	True	
LIOP_MAN_SEL	BOOL	Select AUTO/MANUAL: 1 = Linking, 0 = Operator active	0	False	
AUT_L	BOOL	Linkable input for AUTO/MANUAL mode, 1 = AUTO	0	False	
LIOP_EXT_SEL	BOOL	Select External/Internal SP flag: 1 = from Link, 0 = from Operator	0	False	
EXTRNL_L	BOOL	Linkable input to select External/Internal SP flag, 1=Extrnl	0	False	
LIOP_ ISP_SEL	BOOL	Select internal SP: 1 = from Link, 0 = from Operator	0	False	
SP_L	REAL	Linkable input for internal Setpoint	0.0	False	
LIOP_OUT_SEL	BOOL	Select writing to OUT: 1 = from Link, 0 = from Operator	0	False	
OUT_L	REAL	Linkable input to be written to OUT	0.0	False	
LIOP_RATIO_SEL	BOOL	Select desired ratio: 1 = from Link, 0 = from Operator	0	False	
RAT_DES_L	REAL	Linkable input for desired ratio	0.0	False	

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
SP_OUT	REAL	Calculated set point value	0	False	
OUT	REAL	Calculated output variable that has been limited	0	False	
RAT_ACT	REAL	Actual Ratio Value. Quotient of PV and RAT_INP subject to limits	0	True	
DESRAT	REAL	Desired ratio of PV to RAT_INP	0	False	
ALARM	BOOL	Alarm Status	False	False	
ESP	REAL	External set point	0	True	
H_ALM	BOOL	When TRUE, the High alarm is detected and enabled.	False	False	
L_ALM	BOOL	When TRUE, the Low alarm is detected and enabled.	False	False	
HD_ALM	BOOL	When TRUE, the High Deviation alarm is detected and enabled.	False	False	
LD_ALM	BOOL	When TRUE, the Low Deviation alarm is detected and enabled.	False	False	
AD_ALM	BOOL	When TRUE, the Absolute Deviation alarm is detected and enabled.	False	False	
OOR_ALM	BOOL	When TRUE, the Out of Range alarm is detected and enabled.	False	False	
QUAL_ALM	BOOL	When TRUE, the Quality Not Good alarm is detected and enabled.	False	False	
HH_ALM	BOOL	When TRUE, the High High alarm is detected and enabled.	False	False	
LL_ALM	BOOL	When TRUE, the Low Low alarm is detected and enabled.	False	False	
IR_ALM	BOOL	When TRUE, the Increasing Rate of Change alarm is detected and enabled.	False	False	
DR_ALM	BOOL	When TRUE, the Decreasing Rate of Change alarm is detected and enabled.	False	False	
AR_ALM	BOOL	When TRUE, the Absolute Rate of Change alarm is detected and enabled.	False	False	
VL_EUMIN_OUT	REAL	Copy of Engineering unit: Output, Low	0	False	
VL_EUMAX_OUT	REAL	Copy of Engineering unit: Output, High	0	False	

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
AUTO	BOOL	Auto/manual mode control	False	True	
EXT	BOOL	External/internal mode control	False	True	
SP	REAL	Computer set point value	0	True	
RAMP	BOOL	SP ramping on/off mode control	False	True	
R_TIME	TIME	Ramp time	T#0MS	True	0 to 24d20h31m 23s647ms
SPL	REAL	Set point low limit	0	True	
SPH	REAL	Set point high limit	100	True	
R_RATE	REAL	Ramp rate (in engineering units per min)	0	True	
USE_RR	BOOL	Ramp mode control. Use RTIME or RRATE to control ramp	False	True	
PV_TRACK	BOOL	Internal setpoint tracking control. When Track, internal setpoint tracks PV	True	True	
PG	REAL	Proportional Gain	0.001	True	0.001 to 100.0
TI	REAL	Time-Integral Minutes per Reset	4000	True	0.0 to 4000.0
TD	REAL	Time-Derivative Minutes	0	True	0.00 to 100.0
DG	REAL	Derivative Gain	10	True	2.00 to 20.00
DIR	BOOL	Reverse or Direct action	False	True	
ADAPTIVE_PG_INIT	BOOL	Adaptive Gain Init flag	True	False	
PV_EUMIN	REAL	Engineering unit: Input, Low	0	True	
PV_EUMAX	REAL	Engineering unit: Input, High	100	True	
VL_EUMIN	REAL	Engineering unit: Output, Low	0	True	
VL_EUMAX	REAL	Engineering unit: Output, High	100	True	
VL	REAL	Computer Output, typically written to by an operator command	0	True	
VLL	REAL	OUT low limit input	-3.3	True	
VLH	REAL	OUT high limit input	103.3	True	
H_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the High alarm.	100	True	
H_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the High alarm to clear.	0.5	True	
H_DELAYIN	TIME	The amount of time the input must reach or exceed the High limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms

Name	Data Type	Explanation	Initial	OC&M	Valid Values
H_DELAYOUT	TIME	The amount of time the input must be within the High limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
L_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the Low alarm.	0	True	
L_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Low alarm to clear.	0.5	True	
L_DELAYIN	TIME	The amount of time the input must reach or exceed the Low limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
L_DELAYOUT	TIME	The amount of time the input must be within the Low limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
HD_LIM	REAL	The value (in engineering units) to which (PV) – (DEV) is compared for the High Deviation alarm.	0	True	
HD_DB	REAL	The amount of engineering units by which (PV) – (DEV) must be within the limit for the High Deviation alarm to clear.	0.5	True	
HD_DELAYIN	TIME	The amount of time the input must reach or exceed the High Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
HD_DELAYOUT	TIME	The amount of time the input must be within the High Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
LD_LIM	REAL	The value (in engineering units) to which (DEV) - (PV) is compared for the Low Deviation alarm.	0	True	
LD_DB	REAL	The amount of engineering units by which (DEV) - (PV) must be within the limit for the Low Deviation alarm to clear.	0.5	True	
LD_DELAYIN	TIME	The amount of time the input must reach or exceed the Low Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
LD_DELAYOUT	TIME	The amount of time the input must be within the Low Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
AD_LIM	REAL	The value (in engineering units) to which ABS((PV) – (DEV)) is compared for the Absolute Deviation alarm.	100	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
AD_DB	REAL	The amount of engineering units by which ABS((PV) – (DEV)) must be within the limit for the Absolute Deviation alarm to clear.	0.5	True	
AD_DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
AD_DELAYOUT	TIME	The amount of time the input must be within the Absolute Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
HH_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the High High alarm.	0	True	
HH_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the High High alarm to clear.	0.5	True	
HH_DELAYIN	TIME	The amount of time the input must reach or exceed the High High limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
HH_DELAYOUT	TIME	The amount of time the input must be within the High High limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
LL_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the Low Low alarm.	0	True	
LL_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Low Low alarm to clear.	0.5	True	
LL_DELAYIN	TIME	The amount of time the input must reach or exceed the Low Low limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
LL_DELAYOUT	TIME	The amount of time the input must be within the Low Low limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
IR_LIM	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Increasing Rate of Change alarm.	0	True	
IR_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Increasing Rate of Change alarm to clear.	0.5	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
IR_DELAYIN	TIME	The amount of time the input must reach or exceed the Increasing Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
IR_DELAYOUT	TIME	The amount of time the input must be within the Increasing Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
DR_LIM	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Decreasing Rate of Change alarm.	0	True	
DR_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Decreasing Rate of Change alarm to clear.	0.5	True	
DR_DELAYIN	TIME	The amount of time the input must reach or exceed the Decreasing Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
DR_DELAYOUT	TIME	The amount of time the input must be within the Decreasing Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
AR_LIM	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Absolute Rate of Change alarm.	0	True	
AR_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Absolute Rate of Change alarm to clear.	0.5	True	
AR_DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
AR_DELAYOUT	TIME	The amount of time the input must be within the Absolute Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
OOR_DELAYIN	TIME	The amount of time the input must be outside the limits EINH, EINL to trip the Out of Range alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
OOR_DELAYOUT	TIME	The amount of time the input must be within the limits EINH, EINL to cause the Out of Range alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms

Name	Data Type	Explanation	Initial	OC&M	Valid Values
QUAL_DELAYIN	TIME	The amount of time the input quality must be bad to trip the Quality Not Good alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
QUAL_DELAYOUT	TIME	The amount of time the input quality must be good to cause the Quality Not Good alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
DISABLE_ON_OOS	BOOL	When set to TRUE, the block is disabled if (OOS) is also TRUE.	False	False	
RATIO_MIN	REAL	Minimum value of the ratio	0	True	
RATIO_MAX	REAL	Maximum value of the ratio	1	True	
RAT_DES	REAL	Desired ratio of PV to EXT_INP	0	True	

2.18 SECONDARY_AFB

2.18.1 SECONDARY: Secondary Controller

Function Block Name (compiled version of the Chart)	SECONDARY_AFB
Function Block Number	FB610
Chart Name	SECONDARY
See also, Template Chart (usage example)	**PRIMARY_SECONDARY_AT**

Function

The Secondary Controller (SECONDARY_AFB) block is a factory-compiled function block. It is intended for use as an external-set PID controller with tracking setpoint as part of a cascade circuit. The SECONDARY_AFB block is designed to be used with the PRIMARY_AFB block to form a cascade circuit when it is preferable to control the primary and secondary loops separately. Normally, the output from the primary loop is to be the setpoint to the secondary loop, and the cascade status of the secondary loop is to be fed to the primary loop. High and low alarms are provided for the process variable, and an alarm is provided for absolute deviation between process variable and setpoint.

SECONDARY_AFB is a compiled version of the SECONDARY source chart. Within the chart, there is a control loop that contains SETPNT_AFB, PID_AFB, and AM_AFB blocks along with alarming logic.

The SETPOINT block supplies the setpoint [SP] to the CONTROLLER block. The SETPOINT block has a ramping feature for internal mode (EXT=FALSE). In external mode (EXT=TRUE), the external setpoint [ESP] becomes the controller setpoint, and the SETPOINT block internal setpoint ISP tracks the external setpoint, thereby ensuring no change in setpoint during transition to internal mode. The setpoint tracks the process variable [PV] in manual mode (AUTO=FALSE).

The CONTROLLER block is a PID_AFB block. It operates in auto mode (AUTO=TRUE) and tracks the downstream output [OUT] in manual mode (AUTO=FALSE). The AUTO_MANUAL block provides manual control of the output. In auto mode, the CONTROLLER block output becomes the loop output [OUT]. In manual mode, the AUTO_MANUAL block internal output IOUT becomes the loop output.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32). Additionally, it is called in OB100 and in OB102 (see Start-up Characteristics).

Start-up Characteristics

On a warm restart, OB100 is called and parameters retain the values they had immediately before the restart. On a cold restart, OB102 is called and parameters are initialized with the last downloaded values.

The warm and cold startup behavior of AUTO, SP, EXT, and OUT is configurable, allowing the user to override the default PCS 7 behavior described above if necessary. As delivered, AUTO will be forced to "Manual", EXT will be forced to "Internal" and OUT will be forced to 0.0 on a CPU restart. SP will be set in accordance with default PCS 7 behavior.

To force the AUTO parameter to a specified value on a restart, set the RSTRT_AUTO_FLAG to true ("Force AUTO") and configure RSTRT_AUTO_VALUE with the desired forced value ("Auto" or "Manual"). The RSTRT_AUTO_VALUE will be used to initialize AUTO, if and only if the RSTRT_AUTO_FLAG is true. If RSTRT_AUTO_FLAG is set to false ("Default"), the AUTO parameter will be set in accordance with default PCS 7 behavior.

To force the SP parameter to a specified value on a restart, set the RSTRT_SP_FLAG to true ("Force SP") and configure RSTRT_SP_VALUE with the desired forced value. The RSTRT_SP_VALUE will be used to initialize SP, if and only if the RSTRT_SP_FLAG is true and the block restarts in Auto mode (AUTO="Auto"). If RSTRT_SP_FLAG is set to false ("Default"), the SP parameter will be set in accordance with default PCS 7 behavior.

To force the EXT parameter to a specified value on a restart, set the RSTRT_EXT_FLAG to true ("Force EXT") and configure RSTRT_EXT_VALUE with the desired forced value ("Internal or External"). The RSTRT_EXT_VALUE will be used to initialize EXT, if and only if the RSTRT_EXT_FLAG is true. If RSTRT_EXT_FLAG is set to false ("Default"), the EXT parameter will be set in accordance with default PCS 7 behavior. Forcing the EXT parameter to "External" will cause the SP to be set to the ESP, thereby overriding the SP restart parameter settings.

To force the OUT parameter to a specified value on a restart, set the RSTRT_OUT_FLAG to true ("Force OUT") and configure RSTRT_OUT_VALUE with the desired forced value. The RSTRT_OUT_VALUE will be used to initialize OUT, if and only if the RSTRT_OUT_FLAG is true and the block restarts in Manual mode (AUTO = "Manual"). If RSTRT_OUT_FLAG is set to false ("Default"), the OUT parameter will be set in accordance with default PCS 7 behavior.

Warning

Using "Default" for the RSTRT_OUT_FLAG could result in your output device retaining its last operating state after a restart. Similarly, using "Default" for the RSTRT_AUTO_FLAG could result in your process running in automatic mode after a restart. Use "Forced" for the restart flags and configure the associated restart values to ensure a known restart state.

If restart parameter values are changed offline, the configuration must be compiled (changes only) and downloaded (changes only) to the controller in order for the settings to take effect upon restart. If restart parameter values are changed online (via CFC Test mode), the configuration must be compiled (entire program) and downloaded (entire program) to the controller in order for the settings to take effect upon cold restart. This step is not necessary for the online changes to take effect upon a warm restart.

Operator Control and Monitoring (M&C Relevant)

See Faceplate Information

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

ALARM_8P ID	Message #	Message Class	Default Message Text
EV_ID	1	Warning - above	HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID	2	Warning - below	LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID	3	Alarm - above	HIGH_DEV - $$BlockComment$$ - Limit = @6%f@
EV_ID	4	Alarm - below	LOW_DEV - $$BlockComment$$ - Limit = @7%f@
EV_ID	5	Alarm - above	ABSOLUTE_DEV - $$BlockComment$$ - Limit = @8%f@
EV_ID	6	PLC Process Control Message - Failure	OUT_OF_RANGE - $$BlockComment$$
EV_ID	7	PLC Process Control Message - Failure	QUALITY_NOT_GOOD - $$BlockComment$$
EV_ID2	1	Alarm - above	HIGH_ HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID2	2	Alarm - below	LOW_ LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID2	3	Warning - above	INC_ROC - $$BlockComment$$ - Limit = @6%f@
EV_ID2	4	Warning - below	DEC_ROC - $$BlockComment$$ - Limit = @7%f@
EV_ID2	5	Warning - above	ABS_ROC - $$BlockComment$$ - Limit = @8%f@

Assignment of auxiliary process values to the block parameters

ALARM_8P ID	Value	Block Parameter
EV_ID	@1%s@	BA_NA
EV_ID	@2%d@	BA_ID
EV_ID	@3%d@	STEP_NO
EV_ID	@4%f@	H_LIM
EV_ID	@5%f@	L_LIM
EV_ID	@6%f@	HD_LIM
EV_ID	@7%f@	LD_LIM
EV_ID	@8%f@	AD_LIM
EV_ID2	@1%s@	BA_NA
EV_ID2	@2%d@	BA_ID
EV_ID2	@3%d@	STEP_NO
EV_ID2	@4%f@	HH_LIM
EV_ID2	@5%f@	LL_LIM
EV_ID2	@6%f@	IR_LIM
EV_ID2	@7%f@	DR_LIM
EV_ID2	@8%f@	AR_LIM

2.18.1.1 SECONDARY_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
PV	REAL	Process Variable	0	True	
ESP	REAL	External Setpoint (output from primary controller)	0	True	
TSP	REAL	SP target value for SP ramping	0	True	
H_EN	BOOL	Enable High Alarm	True	True	
L_EN	BOOL	Enable Low Alarm	True	True	
HD_EN	BOOL	Enable High Deviation Alarm	False	True	
LD_EN	BOOL	Enable Low Deviation Alarm	False	True	
AD_EN	BOOL	Enable Absolute Deviation Alarm	True	True	
OOR_EN	BOOL	Enable Out of Range Alarm	False	True	
QUAL_EN	BOOL	Enable Quality Not Good Alarm	False	True	
HH_EN	BOOL	Enable High High Alarm	False	True	
LL_EN	BOOL	Enable Low Low Alarm	False	True	
IR_EN	BOOL	Enable Increasing Rate of Change Alarm	False	True	
DR_EN	BOOL	Enable Decreasing Rate of Change Alarm	False	True	
AR_EN	BOOL	Enable Absolute Rate of Change Alarm	False	True	
OOS	BOOL	Out of Service indicator	False	True	
QBAD	BOOL	Bad quality indicator	False	False	
EV_ID	DWORD	Event ID	DW#16 #5	False	
EV_ID2	DWORD	Event ID	DW#16 #6	False	
SAMPLE_T	REAL	Scan Rate. Use only when compiling CFC as block type.	0.1	False	
RSTRT_AUTO_FLAG	BOOL	Default: Retain on WS, Use load value on CS. Force: Use RSTRT_AUTO_VALUE on restart	True	False	
RSTRT_AUTO_VALUE	BOOL	Init restart value for Auto Mode if RSTRT_AUTO_FLAG true	False	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
RSTRT_EXT_FLAG	BOOL	Default: Retain on WS, Use load value on CS. Force: Use RSTRT_EXT_VALUE on restart	True	False	
RSTRT_EXT_VALUE	BOOL	Init restart value for EXTernal if RSTRT_EXT_FLAG true	False	False	
RSTRT_SP_FLAG	BOOL	Default: Retain on WS, Use load value on CS. Force: Use RSTRT_SP_VALUE on restart.	False	False	
RSTRT_SP_VALUE	REAL	Init restart value for SP if RSTRT_SP_FLAG true and Auto mode.	0	False	
RSTRT_OUT_FLAG	BOOL	Default: Retain on WS, Use load value on CS. Force: Use RSTRT_OUT_VALUE on restart.	True	False	
RSTRT_OUT_VALUE	REAL	Init restart value for OUT if RSTRT_OUT_FLAG true and manual mode	0	False	
BA_EN	BOOL	Batch enabled	False	True	
OCCUPIED	BOOL	Batch occupied	False	True	
BA_ID	DWORD	Batch ID number	0	True	
BA_NA	STRING[32]	Batch name	“	True	
STEP_NO	DWORD	Batch step number	0	True	
DecimalPlacesPV	INT	Decimal places displayed in HMI for PV, SP etc.	2	True	
DecimalPlacesOUT	INT	Decimal places displayed in HMI for OUT, OUT Limits	2	True	
ConfirmChange	BOOL	If TRUE changes from HMI must be confirmed	False	True	
LIOP_MAN_SEL	BOOL	Select AUTO/MANUAL: 1 = Linking, 0 = Operator active	0	False	
AUT_L	BOOL	Linkable input for AUTO/MANUAL mode, 1 = AUTO	0	False	
LIOP_ ISP_SEL	BOOL	Select internal SP: 1 = from Link, 0 = from Operator	0	False	
SP_L	REAL	Linkable input for internal Setpoint	0.0	False	
LIOP_OUT_SEL	BOOL	Select writing to OUT: 1 = from Link, 0 = from Operator	0	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
OUT_L	REAL	Linkable input to be written to OUT	0.0	False	

Note

This list differs significantly from the inputs of the SECONDARY_AFB block in APACS+. In APACS+, only the PV and EXT_SP inputs are present. EXT_SP is now ESP.

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
SP_OUT	REAL	Calculated set point value	0	False	
OUT	REAL	Calculated output variable that has been limited	0	False	
ALARM	BOOL	Alarm Status	False	False	
CASC_S	BOOL	Cascade status	True	False	
H_ALM	BOOL	When TRUE, the High alarm is detected and enabled.	False	False	
L_ALM	BOOL	When TRUE, the Low alarm is detected and enabled.	False	False	
HD_ALM	BOOL	When TRUE, the High Deviation alarm is detected and enabled.	False	False	
LD_ALM	BOOL	When TRUE, the Low Deviation alarm is detected and enabled.	False	False	
AD_ALM	BOOL	When TRUE, the Absolute Deviation alarm is detected and enabled.	False	False	
OOR_ALM	BOOL	When TRUE, the Out of Range alarm is detected and enabled.	False	False	
QUAL_ALM	BOOL	When TRUE, the Quality Not Good alarm is detected and enabled.	False	False	
HH_ALM	BOOL	When TRUE, the High High alarm is detected and enabled.	False	False	
LL_ALM	BOOL	When TRUE, the Low Low alarm is detected and enabled.	False	False	
IR_ALM	BOOL	When TRUE, the Increasing Rate of Change alarm is detected and enabled.	False	False	
DR_ALM	BOOL	When TRUE, the Decreasing Rate of Change alarm is detected and enabled.	False	False	
AR_ALM	BOOL	When TRUE, the Absolute Rate of Change alarm is detected and enabled.	False	False	
VL_EUMIN_OUT	REAL	Copy of VL_EUMIN Input	0	False	
VL_EUMAX_OUT	REAL	Copy of VL_EUMAX Input	0	False	

Note

The SP output has been converted to SP_OUT. SP_OUT is used to interconnect the set-point value to other entities on a chart. The OS interaction is handled by the SP InOut parameter.

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
AUTO	BOOL	Auto/manual mode control	False	True	
EXT	BOOL	Select between internal and external setpoint	False	True	
SP	REAL	Computer set point value	0	True	
RAMP	BOOL	SP ramping on/off mode control	False	True	
R_TIME	TIME	Ramp time	T#0MS	True	0 to 24d20h31m 23s647ms
SPL	REAL	Set point low limit	0	True	
SPH	REAL	Set point high limit	100	True	
R_RATE	REAL	Ramp rate (in engineering units per min)	0	True	
USE_RR	BOOL	Ramp mode control. Use RTIME or RRATE to control ramp	False	True	
PV_TRACK	BOOL	Internal setpoint tracking control. When Track, internal setpoint tracks PV	True	True	
PG	REAL	Proportional Gain	0.001	True	0.001 to 100.0
TI	REAL	Time-Integral Minutes per Reset	4000	True	0.0 to 4000.0
TD	REAL	Time-Derivative Minutes	0	True	0.00 to 100.0
DG	REAL	Derivative Gain	10	True	2.00 to 20.00
DIR	BOOL	Reverse or Direct action	False	True	
ADAPTIVE_PG_INIT	BOOL	Adaptive Gain Init flag	True	False	
PV_EUMIN	REAL	Engineering unit: Input, Low	0	True	
PV_EUMAX	REAL	Engineering unit: Input, High	100	True	
VL_EUMIN	REAL	Engineering unit: Output, Low	0	True	
VL_EUMAX	REAL	Engineering unit: Output, High	100	True	
VL	REAL	Computer Output, typically written to by an operator command	0	True	
VLL	REAL	OUT low limit input	-3.3	True	
VLH	REAL	OUT high limit input	103.3	True	
H_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the High alarm.	100	True	
H_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the High alarm to clear.	0.5	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
H_DELAYIN	TIME	The amount of time the input must reach or exceed the High limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
H_DELAYOUT	TIME	The amount of time the input must be within the High limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
L_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the Low alarm.	0	True	
L_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Low alarm to clear.	0.5	True	
L_DELAYIN	TIME	The amount of time the input must reach or exceed the Low limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
L_DELAYOUT	TIME	The amount of time the input must be within the Low limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
HD_LIM	REAL	The value (in engineering units) to which (PV) – (DEV) is compared for the High Deviation alarm.	0	True	
HD_DB	REAL	The amount of engineering units by which (PV) – (DEV) must be within the limit for the High Deviation alarm to clear.	0.5	True	
HD_DELAYIN	TIME	The amount of time the input must reach or exceed the High Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
HD_DELAYOUT	TIME	The amount of time the input must be within the High Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
LD_LIM	REAL	The value (in engineering units) to which (DEV) - (PV) is compared for the Low Deviation alarm.	0	True	
LD_DB	REAL	The amount of engineering units by which (DEV) - (PV) must be within the limit for the Low Deviation alarm to clear.	0.5	True	
LD_DELAYIN	TIME	The amount of time the input must reach or exceed the Low Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms

Name	Data Type	Explanation	Initial	OC&M	Valid Values
LD_DELAYOUT	TIME	The amount of time the input must be within the Low Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
AD_LIM	REAL	The value (in engineering units) to which ABS((PV) – (DEV)) is compared for the Absolute Deviation alarm.	100	True	
AD_DB	REAL	The amount of engineering units by which ABS((PV) – (DEV)) must be within the limit for the Absolute Deviation alarm to clear.	0.5	True	
AD_DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
AD_DELAYOUT	TIME	The amount of time the input must be within the Absolute Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
HH_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the High High alarm.	0	True	
HH_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the High High alarm to clear.	0.5	True	
HH_DELAYIN	TIME	The amount of time the input must reach or exceed the High High limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
HH_DELAYOUT	TIME	The amount of time the input must be within the High High limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
LL_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the Low Low alarm.	0	True	
LL_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Low Low alarm to clear.	0.5	True	
LL_DELAYIN	TIME	The amount of time the input must reach or exceed the Low Low limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
LL_DELAYOUT	TIME	The amount of time the input must be within the Low Low limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms

Name	Data Type	Explanation	Initial	OC&M	Valid Values
IR_LIM	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Increasing Rate of Change alarm.	0	True	
IR_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Increasing Rate of Change alarm to clear.	0.5	True	
IR_DELAYIN	TIME	The amount of time the input must reach or exceed the Increasing Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
IR_DELAYOUT	TIME	The amount of time the input must be within the Increasing Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
DR_LIM	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Decreasing Rate of Change alarm.	0	True	
DR_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Decreasing Rate of Change alarm to clear.	0.5	True	
DR_DELAYIN	TIME	The amount of time the input must reach or exceed the Decreasing Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
DR_DELAYOUT	TIME	The amount of time the input must be within the Decreasing Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
AR_LIM	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Absolute Rate of Change alarm.	0	True	
AR_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Absolute Rate of Change alarm to clear.	0.5	True	
AR_DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms

Name	Data Type	Explanation	Initial	OC&M	Valid Values
AR_DELAYOUT	TIME	The amount of time the input must be within the Absolute Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
OOR_DELAYIN	TIME	The amount of time the input must be outside the limits EINH, EINL to trip the Out of Range alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
OOR_DELAYOUT	TIME	The amount of time the input must be within the limits EINH, EINL to cause the Out of Range alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
QUAL_DELAYIN	TIME	The amount of time the input quality must be bad to trip the Quality Not Good alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
QUAL_DELAYOUT	TIME	The amount of time the input quality must be good to cause the Quality Not Good alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
DISABLE_ON_OOS	BOOL	When set to TRUE, the block is disabled if (OOS) is also TRUE.	False	False	

2.19 SETPNT_AFB

2.19.1 SETPNT_AFB: Setpoint

Function Block Name	SETPNT_AFB
Function Block Number	FB502

Function

The Setpoint Function Block (Setpnt_AFB) manages the setpoint value (process operating value) which can be derived from two user-specified sources:

- Internal setpoint (ISP)
- External setpoint (ESP)

The setpoint (SP) value is limited by both low and high limits (LOLIM and HILIM) to prevent the setpoint from being inadvertently driven beyond a process operating limit. The internal setpoint can be changed via the Computer Setpoint (CSP) or the Setpoint function block itself (in track or ramp mode).

The Setpoint function block may write either the track variable, (PV) (presumably the control loop process variable), or the External Setpoint (ESP) to its internal setpoint, depending on the modes and options selected. The computer setpoint is written to by an operator command. The SP value is assigned to CSP as well. All values are in engineering units.

The Setpoint function block has a ramping feature to create a ramping setpoint. Ramping is allowed only when the AUTO input is TRUE and the EXTRNL input is FALSE. The ramp start-point is the ISP value. The TARGET value is the intended endpoint. Either a ramp time (RTIME) or a ramp rate (RRATE) controls the ramp slope. The USE_RRATE_SW switch determines which value is used: FALSE for RTIME, TRUE for RRATE. When the RAMP value makes a FALSE-to-TRUE transition, the block initializes the ramp mode. As long as the RAMP and AUTO values are TRUE, and the EXTRNL value is FALSE, ramping is in effect. The SETPNT_AFB function block does not ramp while in either manual or external modes. If the block goes into external or manual while it previously had been in automatic and ramping, the ramping mode is turned off. To re-enable ramping, the block has to be placed into internal mode (EXTRNL = FALSE) and automatic mode (AUTO =TRUE), and then placed in the ramp mode (RAMP = TRUE). The SP output value reflects the ramping of the block and the RTIME output value decrements to show the remaining ramp time. The RRATE value must always be entered as positive; ramp direction is based on the relationship of the target and the starting point. The RRATE value is updated if the RTIME value is used, and the RTIME value is updated if the RRATE value is used. When the TARGET value has been reached, the RAMP input is reset.

The HOLD mode is in effect if the RAMP input value changes from TRUE to FALSE while the RTIME value is greater than zero. The HOLD output value = TRUE to indicate this mode. While in the hold mode, the SP output value is held at its last value. After a FALSE-to-TRUE transition of the RAMP input while the RTIME value is greater than zero, but less than its initial value, the ramp mode is continued. The ramp mode can be continued in two different ways. First, if none

of the three ramp values (TARGET, RTIME or RRATE) have changed, the ramp is continued using the same ramp rate as previous to the HOLD mode, and the current ISP as the restart value; however, the remaining ramp time (RTIME) value is recalculated. Secondly, if any of the three ramp values (TARGET, RTIME, or RRATE) have changed, the new inputs are used just as if the ramp mode was being initialized for the first time. To cancel a ramp, first place the block in hold mode (RAMP = FALSE), then set the RTIME value to zero for a time mode ramp, or set the RRATE value to zero for a rate mode ramp.

Note

When a ramp is finished, the RAMP input/output is reset (RAMP = FALSE).

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32). Additionally, it is called in OB100 and in OB102 (see start-up characteristics).

Start-up Characteristics

On a warm restart, OB100 is called and parameters retain the values they had immediately before the restart. On a cold restart, OB102 is called and parameters are initialized with the last downloaded values.

The warm and cold startup behavior of AUTO and SP is configurable, allowing the user to override the default PCS 7 behavior above if necessary. As delivered, AUTO will be forced to "Manual" and EXT will be forced to "Internal" on a CPU restart. SP will be set in accordance with default PCS 7 behavior.

Warning

Using "Default" for the RSTRT_AUTO_FLAG could result in your process running in automatic mode after a restart. Use "Forced" for the restart flags and configure the associated restart values to ensure a known restart state.

To force the AUTO parameter to a specified value on a restart, set the RSTRT_AUTO_FLAG to true ("Force AUTO") and configure RSTRT_AUTO_VALUE with the desired forced value ("Auto" or "Manual"). The RSTRT_AUTO_VALUE will be used to initialize AUTO, if and only if the RSTRT_AUTO_FLAG is true. If RSTRT_AUTO_FLAG is set to false ("Default"), the AUTO parameter will be set in accordance with default PCS 7 behavior.

To force the SP parameter to a specified value on a restart, set the RSTRT_SP_FLAG to true ("Force SP") and configure RSTRT_SP_VALUE with the desired forced value. The RSTRT_SP_VALUE will be used to initialize SP, if and only if the RSTRT_SP_FLAG is true and the block restarts in Auto mode (AUTO="Auto"). If RSTRT_SP_FLAG is set to false ("Default"), the SP parameter will be set in accordance with default PCS 7 behavior.

To force the EXT parameter to a specified value on a restart, set the RSTRT_EXT_FLAG to true ("Force EXT") and configure RSTRT_EXT_VALUE with the desired forced value ("Internal or External"). The RSTRT_EXT_VALUE will be used to initialize EXT, if and only if the RSTRT_EXT_FLAG is true. If

RSTRT_EXT_FLAG is set to false ("Default"), the EXT parameter will be set in accordance with default PCS 7 behavior. Forcing the EXT parameter to "External" will cause the SP to be set to the ESP, thereby overriding the SP restart parameter settings.

If restart parameter values are changed offline, the configuration must be compiled (changes only) and downloaded (changes only) to the controller in order for the settings to take effect upon restart. If restart parameter values are changed online (via CFC Test mode), the configuration must be compiled (entire program) and downloaded (entire program) to the controller in order for the settings to take effect upon cold restart. This step is not necessary for the online changes to take effect upon a warm restart.

Operator Control and Monitoring (M&C Relevant)

There is no associated faceplate for this block.

2.19.1.1 SETPNT_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
PV	REAL	Process Variable	0	False	
ESP	REAL	External Setpoint	0	False	
TARGET	REAL	Target value for setpoint ramping	0	True	
RSTRT_AUTO_FLAG	BOOL	Default: Retain on WS, Use load val on CS. Force: Use RSTRT_AUTO_VALUE on restart	True	False	
RSTRT_AUTO_VALUE	BOOL	Init restart vlaue for Auto Mode if RSTART_AUTO_FLAG true	False	False	
RSTRT_EXT_FLAG	BOOL	Default: Retain on WS, Use load val on CS. Force: Use RESTRT_EXT_VALUE on restart	True	False	
RSTRT_EXT_VALUE	BOOL	Init restart value on restart	False	False	
RSTRT_SP_FLAG	BOOL	Default: Retain on WS, Use load val on CS. Force: Use RSTRT_SP_VALUE on start	False	False	
RSTRT_SP_VALUE	REAL	Init restart value for SP if RSTRT_SP_FLAG true and Auto mode	0	False	

Note

This list differs from the inputs of the SETPNT_AFB block in APACS+ as follows:

- New parameters were added for restart functionality.
- AUTO and RAMP have been moved from input to input/output.

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
SP	REAL	Calculated setpoint value	0	False	
ISP	REAL	Internal setpoint value	0	False	
LL	BOOL	Low Limit	False	False	
HL	BOOL	High Limit	False	False	
HOLD	BOOL	Ramping mode flag (T=ramping, F=held)	False	False	

Note

This list differs from the outputs of the SETPNT_AFB block in APACS+ as follows:

- RTIME has been moved from output to input/output.

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
AUTO	BOOL	Auto/manual mode control	False	False	
EXTRNL	BOOL	External/internal mode control	False	False	
CSP	REAL	Computer Setpoint	0	True	
RAMP	BOOL	Setpoint is ranping when True	False	True	
RTIME	TIME	Ramp Time	0ms	True	
SLIST_SP	REAL	Internal setpoint	0	False	
LOLIM	REAL	Lower allowable limit of setpoint	100.0	True	
HILIM	REAL	Upper allowable limit of setpoint	0	True	
RRATE	REAL	Ramp rate (in eng. Units/min)	0	True	
USE_RRATE_SW	BOOL	Selects Ramp Rate or Ramp Time as the basis for computing ramping speed	False	True	
PVTRK	BOOL	From APACS+ Help File	True	True	
ESPTRK	BOOL	From APACS+ Help File	True	False	
AUTO_RSTRT	BOOL	Loads AUTO Value up on Restart	False	False	

Note

This list differs from the softlist parameters of the SETPNT_AFB block in APACS+ as follows:

- AUTO and RAMP have been moved from input to input/output.
- RTIME has been moved from output to input/output.

2.20 SINGLE_LOOP_AFB

2.20.1 SINGLE_LOOP_AFB

Function Block Name (compiled version of the Chart)	SINGLE_LOOP_AFB
Function Block Number	FB611
Chart Name	SINGLE_LOOP
See also, Template Chart (usage example)	**SINGLE_LOOP_AT**

Function

The Single Loop Controller (SINGLE_LOOP_AFB) block is a factory-compiled function block. It is intended for use as a single-loop PID controller with a tracking setpoint. Both high and low alarms are provided for the process variable. Another alarm is provided to detect an absolute deviation between the process variable and setpoint.

SINGLE_LOOP_AFB is a compiled version of the SINGLE_LOOP source chart. Within the chart, there is a control loop that contains SETPNT_AFB, PID_AFB, and AM_AFB blocks along with alarming logic.

The SETPOINT block supplies the setpoint [SP] to the CONTROLLER block. The SETPOINT block has a ramping feature for auto mode. The setpoint tracks the process variable [PV] in manual mode (AUTO=FALSE). The CONTROLLER block is a PID_AFB block. It operates in auto mode (AUTO=TRUE) and tracks the downstream output [OUT] in manual mode (AUTO=FALSE). The AUTO_MANUAL block provides manual control of the output. In auto mode the CONTROLLER block output becomes the loop output [OUT]. In manual mode the AUTO_MANUAL block internal output IOUT becomes the loop output.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32). Additionally, it is called in OB100 and in OB102 (see Start-up Characteristics).

Start-up Characteristics

On a warm restart, OB100 is called and parameters retain the values they had immediately before the restart. On a cold restart, OB102 is called and parameters are initialized with the last downloaded values.

The warm and cold startup behavior of AUTO, SP, and OUT is configurable, allowing the user to override the default PCS 7 behavior described above if necessary. As delivered, AUTO will be forced to "Manual" and OUT will be forced to 0.0 on a CPU restart. SP will be set in accordance with default PCS 7 behavior.

To force the AUTO parameter to a specified value on a restart, set the RSTRT_AUTO_FLAG to true ("Force AUTO") and configure

RSTRT_AUTO_VALUE with the desired forced value ("Auto" or "Manual"). The RSTRT_AUTO_VALUE will be used to initialize AUTO, if and only if the RSTRT_AUTO_FLAG is true. If RSTRT_AUTO_FLAG is set to false ("Default"), the AUTO parameter will be set in accordance with default PCS 7 behavior.

To force the SP parameter to a specified value on a restart, set the RSTRT_SP_FLAG to true ("Force SP") and configure RSTRT_SP_VALUE with the desired forced value. The RSTRT_SP_VALUE will be used to initialize SP, if and only if the RSTRT_SP_FLAG is true and the block restarts in Auto mode (AUTO="Auto"). If RSTRT_SP_FLAG is set to false ("Default"), the SP parameter will be set in accordance with default PCS 7 behavior.

To force the OUT parameter to a specified value on a restart, set the RSTRT_OUT_FLAG to true ("Force OUT") and configure RSTRT_OUT_VALUE with the desired forced value. The RSTRT_OUT_VALUE will be used to initialize OUT, if and only if the RSTRT_OUT_FLAG is true and the block restarts in Manual mode (AUTO = "Manual"). If RSTRT_OUT_FLAG is set to false ("Default"), the OUT parameter will be set in accordance with default PCS 7 behavior.

Warning

Using "Default" for the RSTRT_OUT_FLAG could result in your output device retaining its last operating state after a restart. Similarly, using "Default" for the RSTRT_AUTO_FLAG could result in your process running in automatic mode after a restart. Use "Forced" for the restart flags and configure the associated restart values to ensure a known restart state.

If restart parameter values are changed offline, the configuration must be compiled (changes only) and downloaded (changes only) to the controller in order for the settings to take effect upon restart. If restart parameter values are changed online (via CFC Test mode), the configuration must be compiled (entire program) and downloaded (entire program) to the controller in order for the settings to take effect upon cold restart. This step is not necessary for the online changes to take effect upon a warm restart.

Operator Control and Monitoring (M&C Relevant)

See Faceplate Information

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

ALARM_8P ID	Message #	Message Class	Default Message Text
EV_ID	1	Warning - above	HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID	2	Warning - below	LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID	3	Alarm - above	HIGH_DEV - $$BlockComment$$ - Limit = @6%f@
EV_ID	4	Alarm - below	LOW_DEV - $$BlockComment$$ - Limit = @7%f@
EV_ID	5	Alarm - above	ABSOLUTE_DEV - $$BlockComment$$ - Limit = @8%f@
EV_ID	6	PLC Process Control Message - Failure	OUT_OF_RANGE - $$BlockComment$$
EV_ID	7	PLC Process Control Message - Failure	QUALITY_NOT_GOOD - $$BlockComment$$
EV_ID2	1	Alarm - above	HIGH_ HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID2	2	Alarm - below	LOW_ LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID2	3	Warning - above	INC_ROC - $$BlockComment$$ - Limit = @6%f@
EV_ID2	4	Warning - below	DEC_ROC - $$BlockComment$$ - Limit = @7%f@
EV_ID2	5	Warning - above	ABS_ROC - $$BlockComment$$ - Limit = @8%f@

Assignment of auxiliary process values to the block parameters

ALARM_8P ID	Value	Block Parameter
EV_ID	@1%s@	BA_NA
EV_ID	@2%d@	BA_ID
EV_ID	@3%d@	STEP_NO
EV_ID	@4%f@	H_LIM
EV_ID	@5%f@	L_LIM
EV_ID	@6%f@	HD_LIM
EV_ID	@7%f@	LD_LIM
EV_ID	@8%f@	AD_LIM
EV_ID2	@1%s@	BA_NA
EV_ID2	@2%d@	BA_ID
EV_ID2	@3%d@	STEP_NO
EV_ID2	@4%f@	HH_LIM
EV_ID2	@5%f@	LL_LIM
EV_ID2	@6%f@	IR_LIM
EV_ID2	@7%f@	DR_LIM
EV_ID2	@8%f@	AR_LIM

2.20.1.1 SINGLE_LOOP_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
PV	REAL	Process Variable	0	True	
INIT	BOOL	Transitions of the INIT input cause the initialization equation to be performed.	False	False	
TSP	REAL	SP target value for SP ramping	0	True	
H_EN	BOOL	Enable High Alarm	True	True	
L_EN	BOOL	Enable Low Alarm	True	True	
HD_EN	BOOL	Enable High Deviation Alarm	False	True	
LD_EN	BOOL	Enable Low Deviation Alarm	False	True	
AD_EN	BOOL	Enable Absolute Deviation Alarm	True	True	
OOR_EN	BOOL	Enable Out of Range Alarm	False	True	
QUAL_EN	BOOL	Enable Quality Not Good Alarm	False	True	
HH_EN	BOOL	Enable High High Alarm	False	True	
LL_EN	BOOL	Enable Low Low Alarm	False	True	
IR_EN	BOOL	Enable Increasing Rate of Change Alarm	False	True	
DR_EN	BOOL	Enable Decreasing Rate of Change Alarm	False	True	
AR_EN	BOOL	Enable Absolute Rate of Change Alarm	False	True	
OOS	BOOL	Out of Service indicator	False	True	
QBAD	BOOL	Bad quality indicator	False	False	
EV_ID	DWORD	Event ID	Compiler generated for the type or N/A if used as chart *	False	
EV_ID2	DWORD	Event ID	Compiler generated for the type or N/A if used as chart*	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
SAMPLE_T	REAL	Current scan rate supplied by OS	Compiler generated for the type or N/A if used as chart*	False	
RSTRT_AUTO_FLAG	BOOL	Default: Retain on WS, Use load val on CS. Force: Use RSTRT_AUTO_VALVE on restart	True	False	
RSTRT_AUTO_VALVE	BOOL	Init restart value for Auto Mode if RSTRT_AUTO_FLAG	False	False	
RSTRT_SP_FLAG	BOOL	Default: Retain on WS, Use load val on CS. Force: Use RSTRT_SP_VALUE on restart.	False	False	
RSTRT_SP_VALUE	REAL	Init restart value for SP if RSTRT_SP_FLAG true and Auto mode.	0	False	
RSTRT_OUT_FLAG	BOOL	Default: Retain on WS, Use load val on CS. Force: Use RSTRT_OUT_VALUE on restart	True	False	
RSTRT_OUT_VALUE	REAL	Init restart value for OUT if RSTRT_OUT_FLAG true and manual mode.	0	False	
BA_EN	BOOL	Batch enabled	False	True	
OCCUPIED	BOOL	Batch occupied	False	True	
BA_ID	DWORD	Batch ID number	0	True	
BA_NA	STRING[32]	Batch name	‘‘	True	
STEP_NO	DWORD	Batch step number	0	True	
DecimalPlacesPV	INT	Decimal places displayed in HMI for PV, SP etc.	2	True	
DecimalPlacesOUT	INT	Decimal places displayed in HMI for OUT, OUT Limits	2	True	
ConfirmChange	BOOL	If TRUE changes from HMI must be confirmed	False	True	
LIOP_MAN_SEL	BOOL	Select AUTO/MANUAL: 1 = Linking, 0 = Operator active	0	False	
AUT_L	BOOL	Linkable input for AUTO/MANUAL mode, 1 = AUTO	0	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
LIOP_ ISP_SEL	BOOL	Select internal SP: 1 = from Link, 0 = from Operator	0	False	
SP_L	REAL	Linkable input for internal Setpoint	0.0	False	
LIOP_OUT_SEL	BOOL	Select writing to OUT: 1 = from Link, 0 = from Operator	0	False	
OUT_L	REAL	Linkable input to be written to OUT	0.0	False	

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
SP_OUT	REAL	Calculated set point value	0	False	
OUT	REAL	Calculated output variable that has been limited	0	False	
ALARM	BOOL	Alarm Status	False	False	
H_ALM	BOOL	When TRUE, the High alarm is detected and enabled.	False	False	
L_ALM	BOOL	When TRUE, the Low alarm is detected and enabled.	False	False	
HD_ALM	BOOL	When TRUE, the High Deviation alarm is detected and enabled.	False	False	
LD_ALM	BOOL	When TRUE, the Low Deviation alarm is detected and enabled.	False	False	
AD_ALM	BOOL	When TRUE, the Absolute Deviation alarm is detected and enabled.	False	False	
OOR_ALM	BOOL	When TRUE, the Out of Range alarm is detected and enabled.	False	False	
QUAL_ALM	BOOL	When TRUE, the Quality Not Good alarm is detected and enabled.	False	False	
HH_ALM	BOOL	When TRUE, the High High alarm is detected and enabled.	False	False	
LL_ALM	BOOL	When TRUE, the Low Low alarm is detected and enabled.	False	False	
IR_ALM	BOOL	When TRUE, the Increasing Rate of Change alarm is detected and enabled.	False	False	
DR_ALM	BOOL	When TRUE, the Decreasing Rate of Change alarm is detected and enabled.	False	False	
AR_ALM	BOOL	When TRUE, the Absolute Rate of Change alarm is detected and enabled.	False	False	
VL_EUMIN_OUT	REAL	Copy of VL_EUMIN Input	0	True	
VL_EUMAX_OUT	REAL	Copy of VL_EUMAX Input	0	True	

Note

The APACS+ output was renamed to SP_OUT to avoid conflict with the newly added SP In/Out parameter. SP_OUT is used to interconnect the set-point value to other entities on a chart. The OS interaction is handled by the SP In/Out parameter.

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
AUTO	BOOL	Auto/manual mode control	False	True	
SP	REAL	Computer set point value	0	True	
RAMP	BOOL	SP ramping on/off mode control	False	True	
R_TIME	TIME	Ramp time	T#0MS	True	0 to 24d20h31m 23s647ms
SPL	REAL	Set point low limit	0	True	
SPH	REAL	Set point high limit	100	True	
R_RATE	REAL	Ramp rate (in engineering units per min)	0	True	
USE_RR	BOOL	Ramp mode control. Use RTIME or RRATE to control ramp	False	True	
PV_TRACK	BOOL	Internal setpoint tracking control. When Track, internal setpoint tracks PV	True	True	
PG	REAL	Proportional Gain	0.001	True	0.001 to 100.0
TI	REAL	Time-Integral Minutes per Reset	4000	True	0.0 to 4000.0
TD	REAL	Time-Derivative Minutes	0	True	0.00 to 100.0
DG	REAL	Derivative Gain	10	True	2.00 to 20.00
ADAPTIVE_PG_INIT	BOOL	Adaptive Gain Init flag	True	False	
DIR	BOOL	Reverse or Direct action	False	True	
PV_EUMIN	REAL	Engineering unit: Input, Low	0	True	
PV_EUMAX	REAL	Engineering unit: Input, High	100	True	
VL_EUMIN	REAL	Engineering unit: Output, Low	0	True	
VL_EUMAX	REAL	Engineering unit: Output, High	100	True	
VL	REAL	Computer Output, typically written to by an operator command	0	True	
VLL	REAL	OUT low limit input	-3.3	True	
VLH	REAL	OUT high limit input	103.3	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
H_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the High alarm.	100	True	
H_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the High alarm to clear.	0.5	True	
H_DELAYIN	TIME	The amount of time the input must reach or exceed the High limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
H_DELAYOUT	TIME	The amount of time the input must be within the High limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
L_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the Low alarm.	0	True	
L_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Low alarm to clear.	0.5	True	
L_DELAYIN	TIME	The amount of time the input must reach or exceed the Low limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
L_DELAYOUT	TIME	The amount of time the input must be within the Low limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
HD_LIM	REAL	The value (in engineering units) to which (PV) – (DEV) is compared for the High Deviation alarm.	0	True	
HD_DB	REAL	The amount of engineering units by which (PV) – (DEV) must be within the limit for the High Deviation alarm to clear.	0.5	True	
HD_DELAYIN	TIME	The amount of time the input must reach or exceed the High Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
HD_DELAYOUT	TIME	The amount of time the input must be within the High Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
LD_LIM	REAL	The value (in engineering units) to which (DEV) - (PV) is compared for the Low Deviation alarm.	0	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
LD_DB	REAL	The amount of engineering units by which (DEV) - (PV) must be within the limit for the Low Deviation alarm to clear.	0.5	True	
LD_DELAYIN	TIME	The amount of time the input must reach or exceed the Low Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
LD_DELAYOUT	TIME	The amount of time the input must be within the Low Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
AD_LIM	REAL	The value (in engineering units) to which ABS((PV) – (DEV)) is compared for the Absolute Deviation alarm.	100	True	
AD_DB	REAL	The amount of engineering units by which ABS((PV) – (DEV)) must be within the limit for the Absolute Deviation alarm to clear.	0.5	True	
AD_DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
AD_DELAYOUT	TIME	The amount of time the input must be within the Absolute Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
OOR_DELAYIN	TIME	The amount of time the input must be outside the limits EINH, EINL to trip the Out of Range alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
OOR_DELAYOUT	TIME	The amount of time the input must be within the limits EINH, EINL to cause the Out of Range alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
QUAL_DELAYIN	TIME	The amount of time the input quality must be bad to trip the Quality Not Good alarm.	T#0MS	False	0 to 24d20h31m 23s647ms
QUAL_DELAYOUT	TIME	The amount of time the input quality must be good to cause the Quality Not Good alarm to clear.	T#0MS	False	0 to 24d20h31m 23s647ms
DISABLE_ON_OOS	BOOL	When set to TRUE, the block is disabled if (OOS) is also TRUE.	False	False	

2.21 SINGLE_LP_SS_AFB

2.21.1 SINGLE_LP_SS: Single Loop Controller with Standby Sync

Function Block Name (compiled version of the Chart)	SINGLE_LP_SS_AFB
Function Block Number	FB612
Chart Name	SINGLE_LP_SS
See also, Template Chart (usage example)	**SINGLE_LP_SS_AT**

Function

The Single Loop Controller with Standby Sync (SINGLE_LP_SS_AFB) block is a factory-compiled function block. It is intended for use as a single-loop PID controller with tracking setpoint and standby sync functionality. The External Value and Standby Sync command inputs allow the controller output to be immediately forced to any desired value. High and low alarms are provided for the process variable, and an additional alarm is provided to detect an absolute deviation between the process variable and setpoint.

SINGLE_LP_SS_AFB is a compiled version of the SINGLE_LP_SS source chart. Within the chart, there is a control loop that contains SETPNT_AFB, PID_AFB, and AM_AFB blocks along with alarming logic.

The SETPOINT block supplies the setpoint [SP] to the CONTROLLER block. The SETPOINT block has a ramping feature for auto mode. The setpoint tracks the process variable [PV] in manual or standby sync mode (AUTO=FALSE or SS=TRUE). The CONTROLLER block is a PID_AFB block; it operates when AUTO=TRUE and SS=FALSE and tracks the downstream output [OUT] in manual or standby sync mode. The AUTO_MANUAL block provides standby sync and manual control of the output. In standby sync mode, the external value [EXTV] becomes the loop output [OUT]. In auto mode, the CONTROLLER block output becomes the loop output. In manual mode, the AUTO_MANUAL block internal output (IOUT) becomes the loop output.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32). Additionally, it is called in OB100 and in OB102 (see Start-up Characteristics).

Start-up Characteristics

On a warm restart, OB100 is called and parameters retain the values they had immediately before the restart. On a cold restart, OB102 is called and parameters are initialized with the last downloaded values.

The warm and cold startup behavior of AUTO, SP, and OUT is configurable, allowing the user to override the default PCS 7 behavior described above, if

necessary. As delivered, AUTO will be forced to "Manual" and OUT will be forced to 0.0 on a CPU restart. SP will be set in accordance with default PCS 7 behavior.

To force the AUTO parameter to a specified value on a restart, set the RSTRT_AUTO_FLAG to true ("Force AUTO") and configure RSTRT_AUTO_VALUE with the desired forced value ("Auto" or "Manual"). The RSTRT_AUTO_VALUE will be used to initialize AUTO, if and only if the RSTRT_AUTO_FLAG is true. If RSTRT_AUTO_FLAG is set to false ("Default"), the AUTO parameter will be set in accordance with default PCS 7 behavior.

To force the SP parameter to a specified value on a restart, set the RSTRT_SP_FLAG to true ("Force SP") and configure RSTRT_SP_VALUE with the desired forced value. The RSTRT_SP_VALUE will be used to initialize SP, if the RSTRT_SP_FLAG is true and either

- The block is forced to start in Auto mode, or
- AUTO is not forced and the block restarts with Auto mode (AUTO="Auto") and not in standby sync (SS = "False").

If RSTRT_SP_FLAG is set to false ("Default"), the SP parameter will be set in accordance with default PCS 7 behavior.

To force the OUT parameter to a specified value on a restart, set the RSTRT_OUT_FLAG to true ("Force OUT") and configure RSTRT_OUT_VALUE with the desired forced value. The RSTRT_OUT_VALUE will be used to initialize OUT, if and only if the RSTRT_OUT_FLAG is true and the block restarts in Manual mode (AUTO = "Manual"). If RSTRT_OUT_FLAG is set to false ("Default"), the OUT parameter will be set in accordance with default PCS 7 behavior.

Warning

Using "Default" for the RSTRT_OUT_FLAG could result in your output device retaining its last operating state after a restart. Similarly, using "Default" for the RSTRT_AUTO_FLAG could result in your process running in automatic mode after a restart. Use "Forced" for the restart flags and configure the associated restart values to ensure a known restart state.

If restart parameter values are changed offline, the configuration must be compiled (changes only) and downloaded (changes only) to the controller in order for the settings to take effect upon restart. If restart parameter values are changed online (via CFC Test mode), the configuration must be compiled (entire program) and downloaded (entire program) to the controller in order for the settings to take effect upon cold restart. This step is not necessary for the online changes to take effect upon a warm restart.

Operator Control and Monitoring (M&C Relevant)

See Faceplate Information

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

ALARM_8P ID	Message #	Message Class	Default Message Text
EV_ID	1	Warning - above	HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID	2	Warning - below	LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID	3	Alarm - above	HIGH_DEV - $$BlockComment$$ - Limit = @6%f@
EV_ID	4	Alarm - below	LOW_DEV - $$BlockComment$$ - Limit = @7%f@
EV_ID	5	Alarm - above	ABSOLUTE_DEV - $$BlockComment$$ - Limit = @8%f@
EV_ID	6	PLC Process Control Message - Failure	OUT_OF_RANGE - $$BlockComment$$
EV_ID	7	PLC Process Control Message - Failure	QUALITY_NOT_GOOD - $$BlockComment$$
EV_ID2	1	Alarm - above	HIGH_ HIGH - $$BlockComment$$ - Limit = @4%f@
EV_ID2	2	Alarm - below	LOW_ LOW - $$BlockComment$$ - Limit = @5%f@
EV_ID2	3	Warning - above	INC_ROC - $$BlockComment$$ - Limit = @6%f@
EV_ID2	4	Warning - below	DEC_ROC - $$BlockComment$$ - Limit = @7%f@
EV_ID2	5	Warning - above	ABS_ROC - $$BlockComment$$ - Limit = @8%f@

Assignment of auxiliary process values to the block parameters

ALARM_8P ID	Value	Block Parameter
EV_ID	@1%s@	BA_NA
EV_ID	@2%d@	BA_ID
EV_ID	@3%d@	STEP_NO
EV_ID	@4%f@	H_LIM
EV_ID	@5%f@	L_LIM
EV_ID	@6%f@	HD_LIM
EV_ID	@7%f@	LD_LIM
EV_ID	@8%f@	AD_LIM
EV_ID2	@1%s@	BA_NA
EV_ID2	@2%d@	BA_ID
EV_ID2	@3%d@	STEP_NO
EV_ID2	@4%f@	HH_LIM
EV_ID2	@5%f@	LL_LIM

ALARM_8P ID	Value	Block Parameter
EV_ID2	@6%f@	IR_LIM
EV_ID2	@7%f@	DR_LIM
EV_ID2	@8%f@	AR_LIM

2.21.1.1 SINGLE_LP_SS_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
PV	REAL	Process Variable	0	True	
EXTV	REAL	External Variable input	0	False	
SS	BOOL	Standby Sync Command	False	True	
INIT	BOOL	Transitions of the INIT input cause the initialization equation to be performed.	False	False	
TSP	REAL	SP target value for SP ramping	0	True	
H_EN	BOOL	Enable High Alarm	True	True	
L_EN	BOOL	Enable Low Alarm	True	True	
HD_EN	BOOL	Enable High Deviation Alarm	False	True	
LD_EN	BOOL	Enable Low Deviation Alarm	False	True	
AD_EN	BOOL	Enable Absolute Deviation Alarm	True	True	
OOR_EN	BOOL	Enable Out of Range Alarm	False	True	
QUAL_EN	BOOL	Enable Quality Not Good Alarm	False	True	
HH_EN	BOOL	Enable High High Alarm	False	True	
LL_EN	BOOL	Enable Low Low Alarm	False	True	
IR_EN	BOOL	Enable Increasing Rate of Change Alarm	False	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
DR_EN	BOOL	Enable Decreasing Rate of Change Alarm	False	True	
AR_EN	BOOL	Enable Absolute Rate of Change Alarm	False	True	
OOS	BOOL	Out of Service indicator	False	True	
QBAD	BOOL	Bad quality indicator	False	False	
EV_ID	DWORD	Event ID	Compiler generated for the type or N/A if used as chart *	False	
EV_ID2	DWORD	Event ID	Compiler generated for the type or N/A if used as chart *	False	
SAMPLE_T	REAL	Current scan rate supplied by OS	Compiler generated for the type or N/A if used as chart *	False	
RSTRT_AUTO_FLAG	BOOL	Default: Retain on WS, Use load val on CS. Force: Use RSTRT_AUTO_ VALUE	True	False	
RSTRT_AUTO_VALUE	BOOL	Init restart value for Auto Mode if RSTRT_AUTO_ FLAG true	False	False	
RSTRT_SP_FLAG	BOOL	Default: Retain on WS, Use load val on CS. Force: Use RSTRT_SP_VA LUE on restart	False	False	
RSTRT_SP_VALUE	REAL	Init restart value for SP if RSTRT_SP_FL AG true and Auto mode	0	False	
RSTRT_OUT_FLAG	BOOL	Default: Retain on WS, Use load val on CS. Force: Use RSTRT_OUT_V ALUE on restart.	True	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
RSTRT_OUT_VALUE	REAL	Init restart value for OUT if RSTRT_OUT_FLAG true and manual mode	0	False	
BA_EN	BOOL	Batch enabled	False	True	
OCCUPIED	BOOL	Batch occupied	False	True	
BA_ID	DWORD	Batch ID number	0	True	
BA_NA	STRING[32]	Batch name	‘‘	True	
STEP_NO	DWORD	Batch step number	0	True	
DecimalPlacesPV	INT	Decimal places displayed in HMI for PV, SP etc.	2	True	
DecimalPlacesOUT	INT	Decimal places displayed in HMI for OUT, OUT Limits	2	True	
ConfirmChange	BOOL	If TRUE changes from HMI must be confirmed	False	True	
LIOP_MAN_SEL	BOOL	Select AUTO/MANUAL: 1 = Linking, 0 = Operator active	0	False	
AUT_L	BOOL	Linkable input for AUTO/MANUAL mode, 1 = AUTO	0	False	
LIOP_ ISP_SEL	BOOL	Select internal SP: 1 = from Link, 0 = from Operator	0	False	
SP_L	REAL	Linkable input for internal Setpoint	0.0	False	
LIOP_OUT_SEL	BOOL	Select writing to OUT: 1 = from Link, 0 = from Operator	0	False	
OUT_L	REAL	Linkable input to be written to OUT	0.0	False	

Note

This list differs significantly from the inputs of the SINGLE_LP_SS_AFB block in APACS+. The APACS+ list only includes the PV, EXTV, and SS inputs.

*The Update Sampling Time option must be checked when compiling a chart that contains the SINGLE_LP_SS_AFB for this value to be automatically generated by the compiler.

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
SP_OUT	REAL	Calculated set point value	0	False	
OUT	REAL	Calculated output variable that has been limited	0	False	
ALARM	BOOL	Alarm Status	False	False	
H_ALM	BOOL	When TRUE, the High alarm is detected and enabled.	False	False	
L_ALM	BOOL	When TRUE, the Low alarm is detected and enabled.	False	False	
HD_ALM	BOOL	When TRUE, the High Deviation alarm is detected and enabled.	False	False	
LD_ALM	BOOL	When TRUE, the Low Deviation alarm is detected and enabled.	False	False	
AD_ALM	BOOL	When TRUE, the Absolute Deviation alarm is detected and enabled.	False	False	
OOR_ALM	BOOL	When TRUE, the Out of Range alarm is detected and enabled.	False	False	
QUAL_ALM	BOOL	When TRUE, the Quality Not Good alarm is detected and enabled.	False	False	
HH_ALM	BOOL	When TRUE, the High High alarm is detected and enabled.	False	False	
LL_ALM	BOOL	When TRUE, the Low Low alarm is detected and enabled.	False	False	
IR_ALM	BOOL	When TRUE, the Increasing Rate of Change alarm is detected and enabled.	False	False	
DR_ALM	BOOL	When TRUE, the Decreasing Rate of Change alarm is detected and enabled.	False	False	
AR_ALM	BOOL	When TRUE, the Absolute Rate of Change alarm is detected and enabled.	False	False	
VL_EUMIN_OUT	REAL	Copy of Engineering unit: Output, Low	0	False	
VL_EUMAX_OUT	REAL	Copy of Engineering unit: Output, High	0	False	

Note

The APACS+ SP output was renamed to SP_OUT to avoid conflict with the newly added SP In/Out parameter. SP_OUT is used to interconnect the set-point value to other entities on a chart. The OS interaction is handled by the SP In/Out parameter.

Input/ouput

Name	Data Type	Explanation	Initial	OC&M	Valid Values
AUTO	BOOL	Auto/manual mode control	False	True	
SP	REAL	Computer set point value	0	True	
RAMP	BOOL	SP ramping on/off mode control	False	True	
R_TIME	TIME	Ramp time	T#0MS	True	0 to 24d20h31m23s647ms
SPL	REAL	Set point low limit	0	True	
SPH	REAL	Set point high limit	100	True	
R_RATE	REAL	Ramp rate (in engineering units per min)	0	True	
USE_RR	BOOL	Ramp mode control. Use RTIME or RRATE to control ramp	False	True	
PV_TRACK	BOOL	Internal setpoint tracking control. When Track, internal setpoint tracks PV	True	True	
PG	REAL	Proportional Gain	0.001	True	0.001 to 100.0
TI	REAL	Time-Integral Minutes per Reset	4000	True	0.0 to 4000.0
TD	REAL	Time-Derivative Minutes	0	True	0.00 to 100.0
DG	REAL	Derivative Gain	10	True	2.00 to 20.00
DIR	BOOL	Reverse or Direct action	False	True	
ADAPTIVE_PG_INIT	BOOL	Adaptive Gain Init flag	True		
PV_EUMIN	REAL	Engineering unit: Input, Low	0	True	
PV_EUMAX	REAL	Engineering unit: Input, High	100	True	
VL_EUMIN	REAL	Engineering unit: Output, Low	0	True	
VL_EUMAX	REAL	Engineering unit: Output, High	100	True	
VL	REAL	Computer Output, typically written to by an operator command	0	True	
VLL	REAL	OUT low limit input	-3.3	True	
VLH	REAL	OUT high limit input	103.3	True	
H_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the High alarm.	100	True	
H_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the High alarm to clear.	0.5	True	
H_DELAYIN	TIME	The amount of time the input must reach or exceed the High limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms

Name	Data Type	Explanation	Initial	OC&M	Valid Values
H_DELAYOUT	TIME	The amount of time the input must be within the High limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
L_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the Low alarm.	0	True	
L_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Low alarm to clear.	0.5	True	
L_DELAYIN	TIME	The amount of time the input must reach or exceed the Low limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
L_DELAYOUT	TIME	The amount of time the input must be within the Low limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
HD_LIM	REAL	The value (in engineering units) to which (PV) – (DEV) is compared for the High Deviation alarm.	0	True	
HD_DB	REAL	The amount of engineering units by which (PV) – (DEV) must be within the limit for the High Deviation alarm to clear.	0.5	True	
HD_DELAYIN	TIME	The amount of time the input must reach or exceed the High Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
HD_DELAYOUT	TIME	The amount of time the input must be within the High Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
LD_LIM	REAL	The value (in engineering units) to which (DEV) - (PV) is compared for the Low Deviation alarm.	0	True	
LD_DB	REAL	The amount of engineering units by which (DEV) - (PV) must be within the limit for the Low Deviation alarm to clear.	0.5	True	
LD_DELAYIN	TIME	The amount of time the input must reach or exceed the Low Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
LD_DELAYOUT	TIME	The amount of time the input must be within the Low Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms

Name	Data Type	Explanation	Initial	OC&M	Valid Values
AD_LIM	REAL	The value (in engineering units) to which ABS((PV) – (DEV)) is compared for the Absolute Deviation alarm.	100	True	
AD_DB	REAL	The amount of engineering units by which ABS((PV) – (DEV)) must be within the limit for the Absolute Deviation alarm to clear.	0.5	True	
AD_DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Deviation limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
AD_DELAYOUT	TIME	The amount of time the input must be within the Absolute Deviation limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
HH_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the High High alarm.	0	True	
HH_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the High High alarm to clear.	0.5	True	
HH_DELAYIN	TIME	The amount of time the input must reach or exceed the High High limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
HH_DELAYOUT	TIME	The amount of time the input must be within the High High limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
LL_LIM	REAL	The value (in engineering units) to which the (PV) input is compared for the Low Low alarm.	0	True	
LL_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Low Low alarm to clear.	0.5	True	
LL_DELAYIN	TIME	The amount of time the input must reach or exceed the Low Low limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
LL_DELAYOUT	TIME	The amount of time the input must be within the Low Low limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
IR_LIM	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Increasing Rate of Change alarm.	0	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
IR_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Increasing Rate of Change alarm to clear.	0.5	True	
IR_DELAYIN	TIME	The amount of time the input must reach or exceed the Increasing Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
IR_DELAYOUT	TIME	The amount of time the input must be within the Increasing Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
DR_LIM	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Decreasing Rate of Change alarm.	0	True	
DR_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Decreasing Rate of Change alarm to clear.	0.5	True	
DR_DELAYIN	TIME	The amount of time the input must reach or exceed the Decreasing Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
DR_DELAYOUT	TIME	The amount of time the input must be within the Decreasing Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
AR_LIM	REAL	The value (in engineering units per minute) to which the (PV) input is compared for the Absolute Rate of Change alarm.	0	True	
AR_DB	REAL	The amount of engineering units by which (PV) must be within the limit for the Absolute Rate of Change alarm to clear.	0.5	True	
AR_DELAYIN	TIME	The amount of time the input must reach or exceed the Absolute Rate of Change limit to trip an alarm.	T#0MS	False	0 to 24d20h31m23s647ms
AR_DELAYOUT	TIME	The amount of time the input must be within the Absolute Rate of Change limit and deadband to cause the alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms

Name	Data Type	Explanation	Initial	OC&M	Valid Values
OOR_DELAYIN	TIME	The amount of time the input must be outside the limits EINH, EINL to trip the Out of Range alarm.	T#0MS	False	0 to 24d20h31m23s647ms
OOR_DELAYOUT	TIME	The amount of time the input must be within the limits EINH, EINL to cause the Out of Range alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
QUAL_DELAYIN	TIME	The amount of time the input quality must be bad to trip the Quality Not Good alarm.	T#0MS	False	0 to 24d20h31m23s647ms
QUAL_DELAYOUT	TIME	The amount of time the input quality must be good to cause the Quality Not Good alarm to clear.	T#0MS	False	0 to 24d20h31m23s647ms
DISABLE_ON_OOS	BOOL	When set to TRUE, the block is disabled if (OOS) is also TRUE.	False	False	

2.22 VALVE_A_ALM_AFB

2.22.1 VALVE_A_ALM: Valve with Alarm

Function Block Name (compiled version of the Chart)	VALVE_A_ALM_AFB
Function Block Number	FB614
Chart Name	VALVE_A_ALM
See also, Template Chart (usage example)	**VALVE_A_ALM_AT**

Function

This block provides standard valve open/close equipment module functions (similar to the BLK_VLV_1OUT). Device alarming functions are provided by this block.

This factory-compiled block is an encapsulation of the VALVE_A_AFB block with a 4-element discrete alarm block. To provide thorough alarm reporting, the alarm state value of the VALVE_A_AFB block is used to create separate alarms for the following conditions:

- Failure to open (or loss of open verify)
- Failure to close (or loss of close verify)
- Uncommanded close/open or rejected command (due to interlock)
- Failed limit switch (Open and close verify inputs are set)

The latched fail state of the VALVE_A_AFB block is unlatched automatically when the alarm is acknowledged.

The parameters are generally identical to those of the VALVE_A_AFB block. See VALVE_A_AFB for more details and a complete description of the functionality of this block.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32). Additionally, it is called in OB100 and in OB102 (see Start-up Characteristics).

Start-up Characteristics

On a warm restart, OB100 is called and parameters retain the values they had immediately before the restart. On a cold restart, OB102 is called and parameters are initialized with the last downloaded values.

The warm and cold startup behavior of AUTO and OUT is configurable, allowing the user to override the default PCS 7 behavior described above if necessary. As delivered, AUTO will be forced to "Manual" and OUT will be forced to "False" on a CPU restart.

To force the AUTO parameter to a specified value on a restart, set the RSTRT_AUTO_FLAG to true ("Force AUTO") and configure RSTRT_AUTO_VALUE with the desired forced value ("Auto" or "Manual"). The RSTRT_AUTO_VALUE will be used to initialize AUTO, if and only if the RSTRT_AUTO_FLAG is true. If RSTRT_AUTO_FLAG is set to false ("Default"), the AUTO parameter will be set in accordance with default PCS 7 behavior.

To force the OUT parameter to a specified value on a restart, set the RSTRT_OUT_FLAG to true ("Force OUT") and configure RSTRT_OUT_VALUE with the desired forced value ("True" or "False"). The RSTRT_OUT_VALUE will be used to initialize OUT, if and only if the RSTRT_OUT_FLAG is true. If RSTRT_OUT_FLAG is set to false ("Default"), the OUT parameter will be set in accordance with default PCS 7 behavior.

Warning

Using "Default" for the RSTRT_OUT_FLAG could result in your output device retaining its last operating state after a restart. Similarly, using "Default" for the RSTRT_AUTO_FLAG could result in your process running in automatic mode after a restart. Use "Forced" for the restart flags and configure the associated restart values to ensure a known restart state.

If restart parameter values are changed offline, the configuration must be compiled (changes only) and downloaded (changes only) to the controller in order for the settings to take effect upon restart. If restart parameter values are changed online (via CFC Test mode), the configuration must be compiled (entire program) and downloaded (entire program) to the controller in order for the settings to take effect upon cold restart. This step is not necessary for the online changes to take effect upon a warm restart.

Interlock Configuration

All of the Interlocks for this block are visible by default. For each required interlock, configure the following:

- Connect a boolean interlock condition to the Interlock input.
- Configure the description so that it is no longer null ('). If the description is left as a null, the faceplate will not display the status for this interlock. The faceplate is designed to hide an interlock condition when the interlock description is null.

Operator Control and Monitoring (M&C Relevant)

See Faceplate Information

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

Message #	Message Class	Default Message Text
1	PLC Process Control Message - Failure	Failure to open - $$BlockComment$$
2	PLC Process Control Message - Failure	Failure to close - $$BlockComment$$
3	PLC Process Control Message - Failure	Uncommanded close/open or rejected command (due to interlock) - $$BlockComment$$
4	PLC Process Control Message - Failure	Failed limit switch (open and close verify inputs are set) - $$BlockComment$$

Assignment of auxiliary process values to the block parameters

Value	Block Parameter
@1%s@	BA_NA
@2%d@	BA_ID
@3%d@	STEP_NO

2.22.1.1 VALVE_A_ALM_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
OP_VER	BOOL	Open Verification input	False	True	
CL_VER	BOOL	Closed Verification input	False	True	
A_OPEN	BOOL	Automatic Open command input	False	False	
A_CLSE	BOOL	Automatic Close command input	False	False	
PERMIT	BOOL	Permitted to Open input.	True	True	
MODE	BOOL	Mode input from a sequential function chart	False	False	
INT01	BOOL	Interlock 1	True	True	
INT02	BOOL	Interlock 2	True	True	
INT03	BOOL	Interlock 3	True	True	
INT04	BOOL	Interlock 4	True	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
INT05	BOOL	Interlock 5	True	True	
INT06	BOOL	Interlock 6	True	True	
INT07	BOOL	Interlock 7	True	True	
INT08	BOOL	Interlock 8	True	True	
INT09	BOOL	Interlock 9	True	True	
INT10	BOOL	Interlock 10	True	True	
PERMIT_DESC	STRING[32]	Permit Description	''	True	
INT01_DESC	STRING[32]	Interlock 1 Description	''	True	
INT02_DESC	STRING[32]	Interlock 2 Description	''	True	
INT03_DESC	STRING[32]	Interlock 3 Description	''	True	
INT04_DESC	STRING[32]	Interlock 4 Description	''	True	
INT05_DESC	STRING[32]	Interlock 5 Description	''	True	
INT06_DESC	STRING[32]	Interlock 6 Description	''	True	
INT07_DESC	STRING[32]	Interlock 7 Description	''	True	
INT08_DESC	STRING[32]	Interlock 8 Description	''	True	
INT09_DESC	STRING[32]	Interlock 9 Description	''	True	
INT10_DESC	STRING[32]	Interlock 10 Description	''	True	
A1EN	BOOL	Enable Alarm 1	True	True	
A2EN	BOOL	Enable Alarm 2	True	True	
A3EN	BOOL	Enable Alarm 3	True	True	
A4EN	BOOL	Enable Alarm 4	True	True	
OOS	BOOL	Out of Service indicator	False	True	
EV_ID	DWORD	Event ID	Compiler generated for the type or N/A if used as chart *	False	
RSTRT_AUTO_FLAG	BOOL	Default: Retain on WS, Use load val on CS. Force: Use RSTRT_AUTO_VALUE on restart	True	False	
RSTRT_AUTO_VALUE	BOOL	Init restart value for Auto Mode if RSTRT_AUTO_FLAG true	False	False	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
RSTRT_OUT_FLAG	BOOL	Default: Retain on WS, Use load val on CS. Force: Use RSTRT_OUT_VALUE on restart	True	False	
RSTRT_OUT_VALUE	BOOL	Init restart value for OUT if RSTRT_OUT_FLAG	False	False	
BA_EN	BOOL	Batch enabled	False	True	
OCCUPIED	BOOL	Batch occupied	False	True	
BA_ID	DWORD	Batch ID number	0	True	
BA_NA	STRING[32]	Batch name	‘‘	True	
STEP_NO	DWORD	Batch step number	0	True	
ConfirmChange	BOOL	If TRUE changes from HMI must be confirmed	False	True	
LIOP_MAN_SEL	BOOL	Select AUTO/MANUAL: 1 = Linking, 0 = Operator active	False	False	
AUT_L	BOOL	Linkable input for AUTO/MANUAL mode, 1 = AUTO	False	False	

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
OUT	BOOL	Output used to ultimately control the valve	False	True	
INTRLK	BOOL	Interlock OK output	False	True	
READY	BOOL	Ready output	False	False	
STATE	INT	State of the valve	0	True	
ALARM_STATE	INT	Alarm state	0	False	
ALM1	BOOL	Alarm 1	False	False	
ALM2	BOOL	Alarm 2	False	False	
ALM3	BOOL	Alarm 3	False	False	
ALM4	BOOL	Alarm 4	False	False	

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
FAIL	BOOL	Indicates whether valve has a fault condition	False	True	
AUTO	BOOL	Auto mode output	False	True	
OVERRD	BOOL	Override output	False	True	
FAIL_OPEN	BOOL	Indicates fail-safe state of valve	False	True	
OPEN	BOOL	Open command value for manual mode	False	True	
CLOSE	BOOL	Close command value for manual mode	False	True	
OPEN_T	TIME	Maximum time allowed to open the valve	T#10S	True	
CLOS_T	TIME	Maximum time allowed to close the valve	T#10S	True	
AUTO_TRACK	BOOL	If true, block is set to auto mode when SFC transitions to AUTO	False	False	
MODE_HOLD_AUTO	BOOL	Indicates whether block held in auto mod e while SFC is in auto mode	False	False	
ACCEPT_ANY_AUTO	BOOL	When true, block is set to auto when A_O PEN or A_CLSE changes state	False	False	
POS_TRANS_ACLSE	BOOL	Indicates whether a positive transition is required to close the valve	False	False	
AUTO_TRANS_REQ	BOOL	When true, A_OPEN and A_CLSE inputs need to change state to issue open/close cmd	False	False	
FAIL_ON_INTRLK	BOOL	Fail on interlock	True	False	
FAIL_ON_OPEN_CMD	BOOL	Fail on open command	True	False	
A1LIM	BOOL	Alarm 1 Limit	True	True	
A1DELAYIN	TIME	Into Alarm 1 Delay Time	T#0MS	False	0 to 24d20h31m 23s647ms
A1DELAYOUT	TIME	Out of Alarm 1 Delay Time	T#0MS	False	0 to 24d20h31m 23s647ms
A2LIM	BOOL	Alarm 2 Limit	True	True	
A2DELAYIN	TIME	Into Alarm 2 Delay Time	T#0MS	False	0 to 24d20h31m 23s647ms
A2DELAYOUT	TIME	Out of Alarm 2 Delay Time	T#0MS	False	0 to 24d20h31m 23s647ms
A3LIM	BOOL	Alarm 3 Limit	True	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
A3DELAYIN	TIME	Into Alarm 3 Delay Time	T#0MS	False	0 to 24d20h31m 23s647ms
A3DELAYOUT	TIME	Out of Alarm 3 Delay Time	T#0MS	False	0 to 24d20h31m 23s647ms
A4LIM	BOOL	Alarm 4 Limit	True	True	
A4DELAYIN	TIME	Into Alarm 4 Delay Time	T#0MS	False	0 to 24d20h31m 23s647ms
A4DELAYOUT	TIME	Out of Alarm 4 Delay Time	T#0MS	False	0 to 24d20h31m 23s647ms
DISABLE_ON_OOS	BOOL	Disable on Out of Service	False	False	

2.23 VALVE_A_AFB

2.23.1 VALVE_A_AFB: Standard Valve

Function Block Name	VALVE_A_AFB
Function Block Number	FB506

Function

The Standard Valve Function Block (VALVE_A_AFB) provides standard motor start/stop functions that can be easily interfaced to an operator station. This block is based on the APACS+ VALVE_A function block. This block can receive commands from programmed logic, such as a sequential function chart (SFC) while in automatic mode, or it can receive operator-entered commands from an operator interface while in manual mode.

The OP_VER and CL_VER inputs receive the open and close verify signals indicating the state of the valve. A TRUE indicates that the condition is verified. This block accommodates fail-open valves, as well as the normal fail-closed valves. The OUT output value establishes the state of the valve, where TRUE = open and FALSE = close for a fail-close valve, and TRUE = close and FALSE = open for a fail-open valve.

The AUTO parameter indicates the auto/manual state. This state can be changed external to the block. It can also be changed by the block itself, depending on In/Out parameters. When the block is in manual mode (AUTO = FALSE), commands to the block are received via a write of TRUE to the OPEN or CLOSE In/Out parameters. The block automatically resets these values after receiving the command. When the block is in auto mode (AUTO = TRUE), commands to the block are received via the A_OPEN and A_CLSE inputs. Since auto mode is intended for use with commands sent by an SFC, there are some In/Out settings that accommodate this use:

- When AUTO_TRACK is TRUE, the AUTO mode tracks the MODE input (intended to be the mode of the SFC), such that AUTO is set TRUE when MODE becomes TRUE.
- If MODE_HOLD_AUTO is TRUE, AUTO is held TRUE as long as MODE is TRUE. When FALSE, the AUTO value is set when the MODE input value is TRUE, but can then be manually changed.
- If ACCEPT_ANY_AUTO is TRUE, AUTO is set TRUE whenever the A_OPEN (Automatic Open) or A_CLSE (Automatic Close) input changes state.

When in auto mode, the positive transition of the POS_TRANS_ACLSE value determines the function of the A_OPEN and A_CLSE inputs. If the POS_TRANS_ACLSE value is TRUE, the commands need to be FALSE-to-TRUE transitions for both the A_OPEN and A_CLSE inputs. Unlike the OPEN and CLOSE command values, the A_OPEN and A_CLSE input values must be reset external to the block. If the POS_TRANS_ACLSE value is FALSE, the same value should be placed on both the A_OPEN and A_CLSE inputs so that a FALSE-to-TRUE transition sends an open command and a TRUE-to-FALSE transition sends a close command. To avoid the need to generate such a transition for these

commands when the POS_TRANS_ACLSE is FALSE, the AUTO_TRANS_REQ value can be set FALSE. The A_OPEN input (with the same value on the A_CLSE input) is used as the desired state of the valve (TRUE = open, FALSE = closed).

There are many permissive/interlock features built into this block. The PERMIT input, if configured, must be TRUE to permit any open command (or close command if fail-open) to take effect. The block has 10 interlock inputs (INT01 ... INT10), but only the first two are visible by default. All of the configured interlock inputs must be TRUE for the valve to change to the non-fail-safe state; the INTRLK (Interlock) value indicates whether all configured interlock inputs are TRUE. The OVERRD (Override) value represents the override status: when TRUE, any FALSE interlocks (or permit) are overridden, the INTRLK value is set TRUE, and the valve is allowed to leave the fail-safe state. OVERRD may be enabled by a tag write from outside the block. For display at an HMI, there are In/Out values that hold descriptors for the permit (PERMIT) and the interlock (INT01...INT10) inputs.

This block contains failure-checking features that set the FAIL output value to TRUE when a problem is detected. The detected problems include:

- Failure to open before the OPEN_T has expired
- Failure to close before the CLOS_T has expired
- Loss of verify (=UNAVAILABLE data quality or FALSE)
- Uncommanded close (or open for fail open) due to loss of an interlock (enabled by setting the FAIL_ON_INTRLK value to TRUE)
- Rejected open command due to missing permissive or interlock (checking for this failure is always enabled when MODE = TRUE and for all conditions by setting the FAIL_ON_OPEN_CMD value to TRUE.)

The FAIL value is set TRUE when a fail condition is detected. It remains TRUE until it is reset by a tag write from outside of the block (by acknowledge logic or an operator command). The ALARM_STATE value, an integer, is provided to indicate the specific failure condition that occurred.

The READY output indicates that the valve is ready to receive auto commands, where TRUE indicates that the valve is in auto mode and the interlocks are not overridden. The STATE output is an integer value that represents the various states of the valve.

The VALVE_A_AFB supports alarming. An alarm message will be generated if the FAIL parameter becomes TRUE. In APACS+, this functionality was not part of the VALVE_A block, but instead handled by ProcessSuite.

Interlock Configuration

The Interlock parameters are not extensible as they were in *4-mation*. In the PCS 7 implementation of this block, all of the Interlocks exist with only the first two visible by default. If more than two interlocks are required, each additional interlock must be made visible.

For each required interlock, configure the following:

- Connect a boolean interlock condition to the Interlock input.
- Configure the description so that it is no longer null ('). If the description is left as a null, the faceplate will not display the status for this interlock. The faceplate is designed to hide an interlock condition when the interlock description is null.

Calling OBs and Start-up

Calling OBs

This block is called in the watchdog interrupt OB into which the block is installed (for example, OB32). Additionally, it is called in OB100 and in OB102 (see Start-up Characteristics).

Start-up Characteristics

On a warm restart, OB100 is called and parameters retain the values they had immediately before the restart. On a cold restart, OB102 is called and parameters are initialized with the last downloaded values.

The warm and cold startup behavior of AUTO and OUT is configurable, allowing the user to override the default PCS 7 behavior described above if necessary. As delivered, AUTO will be forced to "Manual" and OUT will be forced to "False" on a CPU restart.

To force the AUTO parameter to a specified value on a restart, set the RSTRT_AUTO_FLAG to true ("Force AUTO") and configure RSTRT_AUTO_VALUE with the desired forced value ("Auto" or "Manual"). The RSTRT_AUTO_VALUE will be used to initialize AUTO, if and only if the RSTRT_AUTO_FLAG is true. If RSTRT_AUTO_FLAG is set to false ("Default"), the AUTO parameter will be set in accordance with default PCS 7 behavior.

To force the OUT parameter to a specified value on a restart, set the RSTRT_OUT_FLAG to true ("Force OUT") and configure RSTRT_OUT_VALUE with the desired forced value ("True" or "False"). The RSTRT_OUT_VALUE will be used to initialize OUT, if and only if the RSTRT_OUT_FLAG is true. If RSTRT_OUT_FLAG is set to false ("Default"), the OUT parameter will be set in accordance with default PCS 7 behavior.

Warning

Using "Default" for the RSTRT_OUT_FLAG could result in your output device retaining its last operating state after a restart. Similarly, using "Default" for the RSTRT_AUTO_FLAG could result in your process running in automatic mode after a restart. Use "Forced" for the restart flags and configure the associated restart values to ensure a known restart state.

If restart parameter values are changed offline, the configuration must be compiled (changes only) and downloaded (changes only) to the controller in order for the settings to take effect upon restart. If restart parameter values are changed online (via CFC Test mode), the configuration must be compiled (entire program) and downloaded (entire program) to the controller in order for the settings to take effect upon cold restart. This step is not necessary for the online changes to take effect upon a warm restart.

Operator Control and Monitoring (M&C Relevant)

See Faceplate Information

Message Characteristics

Assignment of message texts and message class to the block parameters

Note

During runtime, the string $$BlockComment$$ will be replaced by the comment for the FB instance that is in alarm.

Message #	Message Class	Default Message Text
1	PLC Process Control Message - Failure	Valve Feedback Failure - $$BlockComment$$

Assignment of auxiliary process values to the block parameters

Value	Block Parameter
@1%s@	BA_NA
@2%d@	BA_ID
@3%d@	STEP_NO

2.23.1.1 VALVE_A_AFB Input Output

Input

Name	Data Type	Explanation	Initial	OC&M	Valid Values
OP_VER	BOOL	Open Verification input	False	True	
CL_VER	BOOL	Closed Verification input	False	True	
A_OPEN	BOOL	Automatic Open command input	False	False	
A_CLSE	BOOL	Automatic Close command input	False	False	
PERMIT	BOOL	Permitted to Open input.	True	True	
MODE	BOOL	Mode input from a sequential function chart	False	False	
INT01	BOOL	Interlock 1	True	True	
INT02	BOOL	Interlock 2	True	True	
INT03	BOOL	Interlock 3	True	True	
INT04	BOOL	Interlock 4	True	True	
INT05	BOOL	Interlock 5	True	True	
INT06	BOOL	Interlock 6	True	True	
INT07	BOOL	Interlock 7	True	True	
INT08	BOOL	Interlock 8	True	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
INT09	BOOL	Interlock 9	True	True	
INT10	BOOL	Interlock 10	True	True	
PERMIT_DESC	STRING[32]	Permit Description	''	True	
INT01_DESC	STRING[32]	Interlock 1 Description	''	True	
INT02_DESC	STRING[32]	Interlock 2 Description	''	True	
INT03_DESC	STRING[32]	Interlock 3 Description	''	True	
INT04_DESC	STRING[32]	Interlock 4 Description	''	True	
INT05_DESC	STRING[32]	Interlock 5 Description	''	True	
INT06_DESC	STRING[32]	Interlock 6 Description	''	True	
INT07_DESC	STRING[32]	Interlock 7 Description	''	True	
INT08_DESC	STRING[32]	Interlock 8 Description	''	True	
INT09_DESC	STRING[32]	Interlock 9 Description	''	True	
INT10_DESC	STRING[32]	Interlock 10 Description	''	True	
RSTRT_AUTO_FLAG	BOOL	Default: Retain on WS, Use load val on CS. Force: Use RSTRT_AUTO_VALUE on restart	True	False	
RSTRT_AUTO_VALUE	BOOL	Init restart value for Auto Mode if RSTRT_AUTO_FLAG true	False	False	
RSTRT_OUT_FLAG	BOOL	Default: Retain on WS, Use load val on CS. Force: Use RSTRT_OUT_VALUE on restart	True	False	
RSTRT_OUT_VALUE	BOOL	Init restart value for OUT if RSTRT_OUT_FLAG true	False	False	
EV_ID	DWORD	Event ID	(Compiler generated)	False	
LIOP_MAN_SEL	BOOL	Select AUTO/MANUAL: 1 = Linking, 0 = Operator active	False	False	
AUT_L	BOOL	Linkable input for AUTO/MANUAL mode, 1 = AUTO	False	False	
BA_EN	BOOL	Batch enabled	False	True	
OCCUPIED	BOOL	Batch occupied	False	True	
BA_ID	DWORD	Batch ID number	0	True	
STEP_NO	DWORD	Batch step number	0	True	
BA_NA	STRING[32]	Batch name	‘‘	True	
ConfirmChange	BOOL	If TRUE changes from HMI must be confirmed	False	True	

Note

This list differs from the inputs of the VALVE_A_AFB block in APACS+ as follows:

- Each input shown above with datatype STRING was a softlist parameter in APACS+.
- The INT01_DESC through INT09_DESC have slightly different names than their APACS+ counterparts. In APACS+ a leading zero was excluded, making the names INT1_DESC through INT9_DESC.
- EV_ID parameter is a new input and is required for the PCS 7 version to support alarming. This parameter did not exist in APACS+.
- Restart parameters (RSTRT_...) did not exist in APACS+.
- LIOP_MAN_SEL and AUT_L added to provide alternate for SET_VAL function.
- Batch parameters BA_EN, OCCUPIED are added to support future batch faceplate view.
- Batch parameters BA_NA, BA_ID, STEP_NO are added to pass batch values in alarm messages.
- ConfirmChange added to give APACS+ users expected feel of "one-step" operator entry

Output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
OUT	BOOL	Output used to ultimately control the valve	False	True	
ALARM	BOOL	Indicates when the Valve is in alarm	False	False	
INTRLK	BOOL	Interlock OK output	False	True	
READY	BOOL	Ready output	False	False	
STATE	INT	State of the valve	0	True	
ALARM_STATE	INT	Alarm state	0	False	

Note

This list differs from the outputs of the VALVE_A_AFB block in APACS+ as follows:

- FAIL and AUTO and OVERRD have been moved to In/Out in PCS 7.
- ALARM_STATE is a new output. In APACS+, it was a softlist parameter.
- ALARM is a new output and is present to support the alarming functionality. This output along with the alarm functionality did not exist in APACS+.
- INTRLK nub is named INT_OK in APACS+. The ProcessSuite name "INTRLK" is preferred.

Integer Values for Valve State

VALUE	STATE	FAIL	OVERRD	OP_VER	CL_VER	INTRLK & PERMIT
0	Undefined	-	-	-	-	-
1	Transition	False	-	False	False	-
2	Closed	False	False	False	True	True
3	Open	False	False	True	False	True
4	Closed and Fail	True	-	False	True	-
5	Open and Fail	True	-	True	False	-
6	Closed and Override	False	True	False	True	-
7	Open and Override	False	True	True	False	-
8	Safe and Interlock not OK	False	False	-	-	False
9	Limit Switch Fail	True	-	True	True	-
10	Transition and Fail	True	-	False	False	-

- means "does not care"

Integer Values for Valve Alarm (Failure) State

VALUE	STATE	STATUS
0	OK	No alarm
1	Failed to Open	OP_VER value not reset in time after OUT value was set TRUE (or FALSE for fail-open).
2	Failed to Close	CL_VER value not reset in time after OUT was set FALSE (or TRUE for fail-open).
3	Loss of Open Verify	OP_VER value went FALSE while open was commanded.
4	Loss of Closed Verify	CL_VER value went FALSE while closed was commanded.
5	Uncommanded Return to Fail-safe	INTRLK value went FALSE while OUT value was TRUE, causing OUT to be SET FALSE.
6	Rejected Command	INTRLK or PERMIT value was FALSE when a command to leave fail-safe state was issued.
7	Failed Limit Switch	OP_VER and CL_VER are both TRUE simultaneously.

Input/output

Name	Data Type	Explanation	Initial	OC&M	Valid Values
FAIL	BOOL	Indicates whether valve has a fault condition	False	True	
AUTO	BOOL	Auto mode output	False	True	
OVERRD	BOOL	Override output	False	True	
AUTO_TRACK	BOOL	If true, block is set to auto mode when SFC transitions to AUTO (MODE becomes "True")	False	False	
MODE_HOLD_AUTO	BOOL	Indicates whether block held in auto mode while SFC is in auto mode	False	False	
ACCEPT_ANY_AUTO	BOOL	When true, block is set to auto when A_OPEN or A_CLSE changes state	False	False	
POS_TRANS_ACLSE	BOOL	Indicates whether a positive transition is required to close the valve	True	False	
AUTO_TRANS_REQ	BOOL	When true, A_OPEN and A_CLSE inputs need to change state to issue open/close cmd	True	False	
FAIL_ON_INTRLK	BOOL	Fail on interlock. When TRUE and the loss of an interlock returns the valve to its fail-safe state (e.g. an uncommanded close), the FAIL status is set TRUE and the ALARM_STATE valve is set to its appropriate value.	False	False	
FAIL_ON_OPEN_CMD	BOOL	Fail on open command. When TRUE and an open command (or a close command for a fail-open valve) is rejected due to the interlocks or permissive, the FAIL status is set TRUE, and the ALARM_STATE value is set to its appropriate value.	False	False	
FAIL_OPEN	BOOL	Indicates fail-safe state of valve	False	True	
OPEN	BOOL	Open command value for manual mode	False	True	
CLOSE	BOOL	Close command value for manual mode	False	True	
OPEN_T	TIME	Maximum time allowed to open the valve	T#10S	True	
CLOS_T	TIME	Maximum time allowed to close the valve	T#10S	True	

Name	Data Type	Explanation	Initial	OC&M	Valid Values
A1EN	BOOL	Alarm 1 Enable	True	True	
A1LIM	BOOL	Alarm 1 Limit	True	False	
A1DELAYIN	TIME	Into Alarm 1 Delay Time	T#0MS	False	0 to 24d20h31m23s647ms
A1DELAYOUT	TIME	Out of Alarm 1 Delay Time	T#0MS	False	0 to 24d20h31m23s647ms
OOS	BOOL	Out of Service indicator	False	True	
DISABLE_ON_OOS	BOOL	Disable on Out of Service	False	False	

Note

This list differs from the softlist parameters of the VALVE_A_AFB block in APACS+ as follows:

- FAIL, AUTO, and OVERRD which were outputs in APACS+, have been moved to In/Out in PCS 7. This is necessary to allow writes from the HMI.
- ALARM_STATE was moved to Output.
- The PT_DESC parameter has been eliminated. Instead, the Comment property of the block has been used to contain the point description information. This is because the faceplate will automatically display the information provided in the Comment field.
- INT01_DESC through INT10_DESC and PERMIT_DESC have been moved to Inputs.
- FAIL_OPEN is named FAIL_OPEN_VALVE in APACS.
- OPEN_T and CLOS_T are named OPEN_TIME and CLOSE_TIME in APACS+.
- The following alarm parameters are new: A1EN, A1LIM, A1DELAYIN, A1DELAYOUT, OOS, DISABLE_ON_OOS. These are needed to support the alarming functionality.

3 Template Charts

A PCS 7 template is to be used in the creation of the controller configuration. Each template contains a function block that has a corresponding symbol and faceplate. The templates provide all the logic required to use the library blocks. All that is necessary on the part of the user is to replace the logical connections with actual addresses. The template charts can be modified or expanded to meet user needs.

PCS 7 **Continuous Function Chart (CFC) Template**
ANALOG_ALM_AT
BLK_VLV_1OUT_AT
BLK_VLV_2OUT_AT
CASCADE_AT
DISCRETE_ALM_AT
EXT_SET_AT
MOTOR_1OUT_AT
MOTOR_2OUT_AT
PRIMARY_SECONDARY_AT
RATIO_SET_AT
SINGLE_LOOP_AT
SINGLE_LP_SS_AT
MOTOR_A_ALM_AT
VALVE_A_ALM_AT

3.1 ANALOG_ALM_AT

This template can be used as an analog indicator with alarms. Both high and low alarms are provided for the process variable. The PV input accepts the process variable and the ALARM output provides the alarm status. The template contains an input driver, along with the ALARM (ANALOG_ALM_AFB) block.

Interconnections

The signal to be monitored must be connected to the VALUE input of the PV_IN block via symbolic addressing.

Simulation

Measured value monitoring can be tested during the configuring phase. This merely requires that input SIM_ON of the PV_IN block be switched on, and that the relevant value be preset at input SIM_V.

Parameter Assignment

To scale the input signal, set the Process Value engineering high via PV_IN's VHRANGE input and the Process Value engineering low via PV_IN's VLRANGE input. These engineering limits will be conveyed to the ALARM's PV_EUMAX and PV_EUMIN, respectively.

By default, the HIGH and LOW alarms are enabled. The various alarms should be enabled/disabled as needed.

Configure the following parameters (which are marked for IEA) in the ALARM for your application:

- All high/low alarm limits (H_LIM, L_LIM)

Faults

Channel faults are detected by the input driver and are assigned to the QBAD input of block ALARM. An alarm message will then be posted if QUAL_EN is enabled.

See also:

ANALOG_ALM_AFB

3.2 BLK_VLV_1OUT_AT

This template can be used as a block valve open/close with one output. It is designed for a normally closed valve with auto/manual operation, pulse commands, and interlocking. An alarm is provided for failure to reach the desired state. The VALVE block has inputs OP_VER, CL_VER, A_OPEN, A_CLSE, and INTRLK for the open verify, close verify, auto open command, auto close command, and interlock; and outputs OUT and FAIL for the output, and failure status of the valve, respectively. The VALVE circuit contains, apart from the VALVE block, two digital input drivers and a digital output driver.

Interconnections

The signal for the open feedback must be connected to the VALUE input of block FB_OPEN via symbolic addressing. The signal for the closed feedback must be connected to the VALUE input of block FB_CLOSED via symbolic addressing. The VALUE output of the OUTPUT block must be connected via symbolic addressing to the signal for of the VALVE to be controlled.

Simulation

If there is no feedback, the simulation can be switched on at driver blocks FB_OPEN and FB_CLOSED (SIM_ON = 1). A connection is provided from the VALVE output OUT to a TIMER_P delay block (SIMULATION) IO input. The SIMULATION QO output is connected to the input SIM_I of block FB_OPEN. It is also connected to input SIM_I of block FB_CLOSED after being inverted. Simulation of valve limit switches is accomplished by delaying the VALVE block's OUT signal. Note than the TIME0 parameter should be less than VALVE's TRAV_T and CLOS_T parameter.

Parameter Assignment

Configure the following parameters for your application:

- To set the VALVE block’s maximum allowed time to open and close, set the parameter TRAV_T of the VALVE block.
- Interlock

Faults

Channel faults are detected by the input and output drivers.

See also:

BLK_VLV_1OUT_AFB

3.3 BLK_VLV_2OUT_AT

This template can be used as a block valve open/close with an open output and a close output. It is designed for a valve with auto/manual operation, pulse commands, and interlocking. An alarm is provided for failure to reach the desired state. The VALVE block has inputs OP_VER, CL_VER, A_OPEN, A_CLSE, and INTRLK for the open verify, close verify, auto open command, auto close command, and interlock; and outputs OP_OUT, CL_OUT, and FAIL for the open output, close output, and failure status of the valve, respectively. The VALVE circuit contains, apart from the VALVE block, two digital input drivers and two digital output drivers.

Interconnections

The signal for the open feedback must be connected to the VALUE input of block FB_OPEN via symbolic addressing. The signal for the closed feedback must be connected to the VALUE input of block FB_CLOSED via symbolic addressing. The VALUE output of the OPEN_OUTPUT block must be connected via symbolic addressing to the open signal for the valve to be controlled. The VALUE output of the CLOSE_OUTPUT block must be connected via symbolic addressing to the close signal for output of the valve to be controlled.

Simulation

If there is no feedback, the simulation can be switched on at driver blocks FB_OPEN and FB_CLOSED (SIM_ON = 1). An SR_FF block (SIMULATION) is used to implement the simulation of the feedback.

Parameter Assignment

Configure the following parameters for your application:

- To set the VALVE block's maximum allowed time to open and close, set the parameter TRAV_T of the VALVE block.
- Interlock

Faults

Channel faults are detected by the input and output drivers.

See also:

BLK_VLV_2OUT_AFB

3.4 CASCADE_AT

This template can be used as a self-contained cascade PID controller with tracking setpoints. The output from the primary control loop is the setpoint to a secondary control loop. High and low alarms are provided for the process variables of both loops. Additional alarms are provided to detect a deviation between the process variable and setpoint of both loops. The block has inputs P_PV and S_PV for the primary and secondary process variables, and outputs P_SP_OUT, S_SP_OUT, OUT, and ALARM for the primary setpoint, secondary setpoint, output, and alarm status of the loops, respectively. The control loop contains, apart from the controller, two input drivers (PRIMARY_PV_IN, SECONDARY_PV_IN), an output driver (SECONDARY_VL_OUT), and delay elements for simulation.

Interconnections

The signal for the primary value must be connected to the VALUE input of block PRIMARY_PV_IN via symbolic addressing. The signal for the secondary actual value must be connected to the VALUE input of block SECONDARY_PV_IN via symbolic addressing. The VALUE output of the SECONDARY_VL_OUT block must be connected via symbolic addressing to the signal for the actuator to be controlled.

Simulation

During the configuring phase, the controller can be tested with the delay element. This merely requires that input SIM_ON of block PRIMARY_PV_IN and the input SIM_ON of block SECONDARY_PV_IN be switched on. The Simulate Secondary CFC implements simple simulation by feeding back a scaled lagged copy of the CONTROLLER OUT signal. Normally, only the TM_LAG value needs to be adjusted for the application. The Simulate Primary CFC implements simple simulation by feeding back a scaled lagged copy of the Simulate Secondary's PV. Normally, only the TM_LAG value needs to be adjusted for the application.

Parameter Assignment

To scale the primary input signal, set the primary's engineering high via PRIMARY_PV_IN's VHRANGE input and the Process Value engineering low via PRIMARY_PV_IN's VLRANGE input.

To scale the secondary input signal, set the secondary's engineering high via SECONDARY_PV_IN's VHRANGE input and the Process Value engineering low via SECONDARY_PV_IN's VLRANGE input.

This chart presumes that the CONTROLLER P_VL_EUMIN should equal the CONTROLLER S_PV_EUMIN and that the CONTROLLER P_VL_EUMAX should equal the CONTROLLER S_PV_EUMAX.

To scale the output signal, set the Control OUT value engineering high via CONTROLLER's S_VL_EUMAX input and the Control OUT Engineering Low via S_VL_EUMIN input. These engineering limits will be conveyed to the S_VL_EUMAX_OUT output and S_VL_EUMIN_OUT outputs, respectively. As configured, SECONDARY_VL_OUT's upper limit (UHRANGE) and the

SECONDARY_VL_OUT's lower limit (ULRANGE) match the CONTROLLER's S_VL_EUMAX and S_VL_EUMIN, respectively.

Configure the following parameters (which are marked for IEA) in the CONTROLLER (both primary and secondary) for your application:

- All tuning parameters (PG, TI, TD and DG)
- All high/low alarm and absolute deviation limits (H_LIM, L_LIM, AD_LIM)
- Valve scaling units (VL_EUMIN, VL_EUMAX)
- Valve "clamping" limits (VLH, VLL). Note that VLH cannot exceed 3% of span above VL_EUMAX and VLL cannot exceed 3% of span below VL_EUMIN.
- Setpoint limits (SPH, SPL)

Faults

Channel faults are detected by the PRIMARY_PV_IN input and are assigned to input P_QBAD of the CONTROLLER block. An alarm message will then be posted if P_QUAL_EN is enabled. Channel faults are detected by the SECONDARY_PV_IN input and the SECONDARY_VL_OUT and are assigned to input S_QBAD of the CONTROLLER block. An alarm message will then be posted if S_QUAL_EN is enabled.

See also:

CASCADE_AFB

3.5 DISCRETE_ALM_AT

This template can be used to implement a discrete indicator with alarm. The template includes a digital input driver. A discrete alarm is provided for the discrete input (DN). The ALARM output provides the alarm status of the block. The block provides an alarm when the discrete input is in an undesired state.

Interconnections

The signal to be monitored must be connected to the VALUE input of the SIGNAL block via symbolic addressing.

Simulation

Discrete value monitoring can be tested during the configuring phase. This merely requires that input SIM_ON of the SIGNAL block be switched on, and that the value 1 or 0 be preset at input SIM_I.

Parameter Assignment

Configure the LIMIT parameter (which is marked for IEA) in the ALARM for your application.

Faults

Channel faults are detected by the input driver block.

See also:

DISCRETE_ALM_AFB

3.6 EXT_SET_AT

This template can be used as an external-set PID controller with tracking setpoint. High and low alarms are provided for the process variable. Another alarm is provided to detect deviation between the process variable and setpoint. The CONTROLLER block has PV and ESP inputs for the process variable and external setpoint. It also has SP_OUT, OUT, and ALARM outputs for the setpoint, output, and alarm status of the loop, respectively. The control loop contains, apart from the controller, an input driver, an output driver, and a delay element for simulation.

Interconnections

The signal for the process value must be connected to the VALUE input of block PV_IN via symbolic addressing. The VALUE output of the OUTPUT block must be connected via symbolic addressing to the signal for the actuator to be controlled.

Simulation

During the configuring phase, the controller can be tested with the delay element. This merely requires that input SIM_ON of block PV_IN be switched on. The Simulation CFC implements simple simulation by feeding back a scaled lagged copy of the CONTROLLER's OUT signal. Normally, only the TM_LAG value needs to be adjusted for the application.

Parameter Assignment

To scale the input signal, set the Process Value engineering high via the input driver PV_IN's VHRANGE input and the Process Value engineering low via PV_IN's VLRANGE input. These engineering limits will be conveyed to the CONTROLLER's PV_EUMAX and PV_EUMIN, respectively.

To scale the output signal, set the Control OUT value engineering high via CONTROLLER's VL_EUMAX input and the Control OUT Engineering Low via CONTROLLER's VL_EUMIN input. These engineering limits will be conveyed to the CONTROLLER's VL_EUMAX_OUT and VL_EUMIN_OUT outputs, respectively. As configured, OUTPUT's upper limit (UHRANGE) and the OUTPUT's lower limit (ULRANGE) match the CONTROLLER's VL_EUMAX and VL_EUMIN, respectively.

Configure the following parameters (which are marked for IEA) for your application:

- All tuning parameters (PG, TI, TD and DG)
- All high/low alarm and absolute deviation limits (H_LIM, L_LIM, AD_LIM)
- Valve scaling units (VL_EUMIN, VL_EUMAX)
- Valve "clamping" limits (VLH, VLL). Note that VLH cannot exceed 3% of span above VL_EUMAX and VLL cannot exceed 3% of span below VL_EUMIN.
- Setpoint limits (SPH, SPL)

Faults

Channel faults are detected by the input and output drivers and are assigned to input QBAD of the CONTROLLER block. An alarm message will then be posted if QUAL_EN is enabled.

See also:

EXT_SET_AFB

3.7 MOTOR_A_ALM_AT

This template can be used to implement standard motor start/stop functions. Device alarming functions are provided. This MOTOR block is an encapsulation of the MOTOR_A block with a 4-element discrete alarm block. The MOTOR block has inputs PROOF, A_STRT, A_STOP, and INT01-INT11 for the proof, auto start command, auto stop command, and eleven interlocks, and parameters OUT, and FAIL for the output, and failure status of the motor, respectively. Alarms are reported for the following conditions:

- Failure to start (or loss of proof)
- Failure to stop (or unexpected proof)
- Uncommanded stop or rejected start command (due to interlock)

The latched fail state is unlatched automatically when the alarm is acknowledged.

The motor circuit contains, apart from the motor block, a digital input driver and digital output driver.

Interconnections

The signal for the feedback run must be connected to the VALUE input of block FB_RUN via symbolic addressing. The VALUE output of the OUTPUT block must be connected via symbolic addressing to the signal for the motor to be controlled.

Simulation

If there is no run feedback, the simulation can be switched on at driver block FB_RUN (SIM_ON = 1). Simulation of motor proof is accomplished by delaying the MOTOR block's OUT signal via a TIMER_P block (SIMULATION). Note that the TIME0 parameter should be less than MOTOR block's STRT_T and STOP_T parameters.

Parameter Assignment

Configure the following parameters for your application:

- To set the MOTOR block's maximum allowed time to start and stop, set the parameter STRT_T and STOP_T of the MOTOR block (which are marked for IEA) individually.
- Interlocks and Interlock descriptions.
- Permit and Permit description.

Faults

Channel faults are detected by the input and output drivers.

See also:

MOTOR_A_ALM_AFB

3.8 MOTOR_1OUT_AT

This template can be used to implement a motor start/stop with one output. The block is designed for a motor with auto/manual operation, pulse commands, and interlocking. An alarm is provided for failure to reach the desired state. The block has inputs PROOF, A_STRT, A_STOP, and INTRLK for the proof, auto start command, auto stop command, and interlock; and outputs OUT, and FAIL for the output, and failure status of the motor, respectively. The motor circuit contains, apart from the motor block, a digital input driver and digital output driver.

Interconnections

The signal for the feedback run must be connected to the VALUE input of block FB_RUN via symbolic addressing. The VALUE output of the OUTPUT block must be connected via symbolic addressing to the signal for the motor to be controlled.

Simulation

If there is no feedback run, the simulation can be switched on at driver block FB_RUN (SIM_ON = 1). Simulation of motor proof is accomplished by delaying the MOTOR block's OUT signal via a TIMER_P block (SIMULATION). Note that the TIME0 parameter should be less than MOTOR block's STRT_T and STOP_T parameters.

Parameter Assignment

Configure the following parameters for your application:

- To set the MOTOR block's maximum allowed time to start and stop, set the parameters STRT_T and STOP_T of the MOTOR block (which are marked for IEA) individually.
- Interlock

Faults

Channel faults are detected by the input and output drivers.

See also:

MOTOR_1OUT_AFB

3.9 MOTOR_2OUT_AT

This template can be used to implement a motor start/stop with a start output and a stop output. The block is designed for a motor with auto/manual operation, pulse commands, and interlocking. An alarm is provided for failure to reach the desired state. The MOTOR block has inputs PROOF, A_STRT, A_STOP, and INTRLK for the proof, auto start command, auto stop command, and interlock; and outputs STRT_O, STOP_O, and FAIL for the start output, stop output, and failure status of the motor, respectively. The motor circuit contains, apart from the motor block, a digital input driver and two digital output drivers.

Interconnections

The signal for the feedback run must be connected to the VALUE input of block FB_RUN via symbolic addressing. The VALUE output of the STRT_OUTPUT block must be connected via symbolic addressing to the signal for the motor to be controlled. The VALUE output of the STOP_OUTPUT block must be connected via symbolic addressing to the signal for the motor to be controlled.

Simulation

If there is no feedback run, the simulation can be switched on at driver block FB_RUN (SIM_ON = 1). An SR_FF block (Simulate State) and a TIMER_P block (Simulate Delay) is used to implement the simulation of the feedback. Note that the TIMEO parameter should be less than MOTOR's STRT_T and STOP_T parameters.

Parameter Assignment

Configure the following parameters for your application:

- To set the MOTOR block's maximum allowed time to start and stop, set the parameter STRT_T and STOP_T of the MOTOR block (which are marked for IEA) individually.
- Interlock

Faults

Channel faults are detected by the input and output drivers.

See also:

MOTOR_2OUT_AFB

3.10 PRIMARY_SECONDARY_AT

This template can be used as a cascade circuit. The output from the primary loop is the setpoint to the secondary loop, and the cascade status of the secondary loop is fed to the primary loop. High and low alarms are provided for each process variable, and an alarm is provided for deviation between process variable and setpoint. The PRIMARY block has inputs PV, FDBK, and CASC_S for the process variable, feedback, and cascade status, and outputs SP_OUT, OUT, and ALARM for the setpoint, output, and alarm status of the loop, respectively. The SECONDARY block has inputs PV and ESP for the process variable and external setpoint, and outputs SP_OUT, OUT, ALARM, and CASC_S for the setpoint, output, alarm status, and cascade status of the loop, respectively. The control loop contains, apart from the controller, two input drivers (PRIMARY_PV_IN, SECONDARY_PV_IN), an output driver (SECONDARY_VL_OUT) and delay elements for simulation.

Interconnections

The signal for the primary value must be connected to the VALUE input of block PRIMARY_PV_IN via symbolic addressing. The signal for the secondary value must be connected to the VALUE input of block SECONDARY_PV_IN via symbolic addressing. The VALUE output of the SECONDARY_VL_OUT block must be connected via symbolic addressing to the signal for the actuator to be controlled.

Simulation

During the configuring phase, the controller can be tested with the delay element. This merely requires that input SIM_ON of block PRIMARY_PV_IN and the input SIM_ON of block SECONDARY_PV_IN be switched on. The Simulate Secondary CFC implements simple simulation by feeding back a scaled lagged copy of the SECONDARY's OUT signal. Normally, only the TM_LAG value needs to be adjusted for the application. The Simulate Primary CFC implements simple simulation by feeding back a scaled lagged copy of Simulate Secondary's PV. Normally, only the TM_LAG value needs to be adjusted for the application.

Parameter Assignment

To scale the primary input signal, set the primary's engineering high via PRIMARY_PV_IN's VHRANGE input and the Process Value engineering low via PRIMARY_PV_IN's VLRANGE input.

To scale the secondary input signal, set the secondary's engineering high via SECONDARY_PV_IN's VHRANGE input and the Process Value engineering low via SECONDARY_PV_IN's VLRANGE input.

This chart presumes that the PRIMARY VL_EUMIN should equal the SECONDARY PV_EUMIN and that the PRIMARY VL_EUMAX should equal the SECONDARY PV_EUMAX.

To scale the output signal, set the Control OUT value engineering high via SECONDARY's VL_EUMAX input and the Control OUT Engineering Low via SECONDARY 's VL_EUMIN input. These engineering limits will be conveyed to

the SECONDARY's VL_EUMAX_OUT and VL_EUMIN_OUT outputs, respectively. As configured, SECONDARY_VL_OUT's upper limit (UHRANGE) and the SECONDARY_VL_OUT's lower limit (ULRANGE) match the SECONDARY's VL_EUMAX and VL_EUMIN, respectively.

Configure the following parameters (which are marked for IEA) in the PRIMARY and SECONDARY for your application:

- All tuning parameters (PG, TI, TD and DG)
- All high/low alarm and absolute deviation limits (H_LIM, L_LIM, AD_LIM)
- Valve scaling units (VL_EUMIN, VL_EUMAX)
- Valve "clamping" limits (VLH, VLL). Note that VLH cannot exceed 3% of span above VL_EUMAX and VLL cannot exceed 3% of span below VL_EUMIN.
- Setpoint limits (SPH, SPL)

Faults

Channel faults are detected by the PRIMARY_PV_IN input and are assigned to input QBAD of the PRIMARY block. An alarm message will then be posted if QUAL_EN is enabled. Channel faults are detected by the SECONDARY_PV_IN input and the SECONDARY_VL_OUT and are assigned to input QBAD of the SECONDARY block. An alarm message will then be posted if QUAL_EN is enabled.

See also:

PRIMARY_AFB

SECONDARY_AFB

3.11 RATIO_SET_AT

This template can be used use as a ratio-set PID controller with tracking setpoint. An external ratio input is multiplied by a desired ratio value (i.e. a controlled variable divided by wild variable) to calculate the external setpoint. The process variable is divided by the external input for display of the actual ratio. Both high and low alarms are provided for the process variable, and an additional alarm is provided to detect a deviation between the process variable and setpoint. The CONTROLLER block has PV and RAT_INP inputs for the process variable and external input. It also has SP_OUT, OUT, RAT_ACT, DESRAT, and ALARM outputs for the setpoint, output, actual ratio, desired ratio, and alarm status of the loop, respectively. The control loop contains, apart from the controller, two input drivers (PV_IN and RATIO_IN), an output driver (OUTPUT) and a delay element for simulation.

Interconnections

The signal for the process value must be connected to the VALUE input of block PV_IN via symbolic addressing. The signal for the external ratio actual value must be connected to the VALUE input of the block RATIO_IN via symbolic addressing. The VALUE output of the OUTPUT block must be connected via symbolic addressing to the signal for the actuator to be controlled.

Simulation

During the configuring phase, the controller can be tested with the delay element. This merely requires that input SIM_ON of block PV_IN be switched on. The Simulation CFC implements simple simulation by feeding back a scaled lagged copy of the CONTROLLER's OUT signal. Normally, only the TM_LAG value needs to be adjusted for the application.

Parameter Assignment

To scale the input signal, set the Process Value engineering high via the input driver PV_IN’s VHRANGE input and the Process Value engineering low via PV_IN's VLRANGE input. These engineering limits will be conveyed to the CONTROLLER's PV_EUMAX and PV_EUMIN, respectively.

To scale the output signal, set the Control OUT value engineering high via CONTROLLER's VL_EUMAX input and the Control OUT Engineering Low via CONTROLLER's VL_EUMIN input. These engineering limits will be conveyed to the CONTROLLER's VL_EUMAX_OUT and VL_EUMIN_OUT outputs, respectively. As configured, the OUTPUT's upper limit (UHRANGE) and the OUTPUT's lower limit (ULRANGE) match the CONTROLLER's VL_EUMAX and VL_EUMIN, respectively

Configure the following parameters (which are marked for IEA) for your application:

- All tuning parameters (PG, TI, TD and DG)
- All high/low alarm and absolute deviation limits (H_LIM, L_LIM, AD_LIM)
- Valve scaling units (VL_EUMIN, VL_EUMAX)

- Valve "clamping" limits (VLH, VLL). Note that VLH cannot exceed 3% of span above VL_EUMAX and VLL cannot exceed 3% of span below VL_EUMIN.
- Setpoint limits (SPH, SPL)

Faults

Channel faults are detected by the input and output drivers and are assigned to input QBAD of the CONTROLLER block. An alarm message will then be posted if QUAL_EN is enabled.

See also:

RATIO_SET_AFB

3.12 SINGLE_LOOP_AT

This template can be used to implement a single-loop PID controller with a tracking setpoint. Both high and low alarms are provided for the process variable. Another alarm is provided to detect a deviation between the process variable and setpoint. The controller block has a PV input for the process variable. It also has SP_OUT, OUT, and ALARM outputs for the setpoint, output, and alarm status of the loop, respectively. The control loop contains, apart from the controller, an input driver (PV_IN), an output driver (OUTPUT) and a delay element for simulation.

Interconnections

The signal for the actual value must be connected to the VALUE input of block PV_IN via symbolic addressing. The VALUE output of the OUTPUT block must be connected via symbolic addressing to the signal for the actuator to be controlled.

Simulation

During the configuring phase, the controller can be tested with the delay element. This merely requires that input SIM_ON of block PV_IN be switched on. The Simulation CFC implements simple simulation by feeding back a scaled lagged copy of the CONTROLLER's OUT signal. Normally, only the TM_LAG value needs to be adjusted for the application.

Parameter Assignment

To scale the input signal, set the Process Value engineering high via the input driver PV_IN's VHRANGE input and the Process Value engineering low via PV_IN's VLRANGE input. These engineering limits will be conveyed to the CONTROLLER's PV_EUMAX and PV_EUMIN, respectively.

To scale the output signal, set the Control OUT value engineering high via CONTROLLER's VL_EUMAX input and the Control OUT Engineering Low via CONTROLLER's VL_EUMIN input. These engineering limits will be conveyed to the CONTROLLER's VL_EUMAX_OUT and VL_EUMIN_OUT outputs, respectively. As configured, OUTPUT's upper limit (UHRANGE) and the OUTPUT's lower limit (ULRANGE) match the CONTROLLER's VL_EUMAX and VL_EUMIN, respectively.

Configure the following parameters (which are marked for IEA) for your application:

- All tuning parameters (PG, TI, TD and DG)
- All high/low alarm and absolute deviation limits (H_LIM, L_LIM, AD_LIM)
- Valve scaling units (VL_EUMIN, VL_EUMAX)
- Valve "clamping" limits (VLH, VLL). Note that VLH cannot exceed 3% of span above VL_EUMAX and VLL cannot exceed 3% of span below VL_EUMIN.
- Setpoint limits (SPH, SPL)

Faults

Channel faults are detected by the input and output drivers and are assigned to input QBAD of the CONTROLLER block. An alarm message will then be posted if QUAL_EN is enabled.

See also:

SINGLE_LOOP_AFB

3.13 SINGLE_LP_SS_AT

This template can be used to implement a single-loop PID controller with tracking setpoint and standby sync functionality. The External Value and Standby Sync command inputs allow the controller output to be immediately forced to any desired value. High and low alarms are provided for the process variable, and an additional alarm is provided to detect a deviation between the process variable and setpoint. The CONTROLLER block has PV, EXTV, and SS inputs for process variable, external value, and standby sync command. It also has SP_OUT, OUT, and ALARM outputs for the setpoint, output, and alarm status of the loop, respectively. The control loop contains, apart from the controller, an input driver (PV_IN), an output driver (OUTPUT) and a delay element for simulation.

Interconnections

The signal for the process value must be connected to the VALUE input of block PV_IN via symbolic addressing. The VALUE output of the OUTPUT block must be connected via symbolic addressing to the signal for the actuator to be controlled.

Simulation

During the configuring phase, the controller can be tested with the delay element. This merely requires that input SIM_ON of block PV_IN be switched on. The Simulation CFC implements simple simulation by feeding back a scaled lagged copy of the CONTROLLER's OUT signal. Normally, only the TM_LAG value needs to be adjusted for the application.

Parameter Assignment

To scale the input signal, set the Process Value engineering high via the input driver PV_IN's VHRANGE input and the Process Value engineering low via PV_IN's VLRANGE input. These engineering limits will be conveyed to the CONTROLLER's PV_EUMAX and PV_EUMIN, respectively.

To scale the output signal, set the Control OUT value engineering high via CONTROLLER's VL_EUMAX input and the Control OUT Engineering Low via CONTROLLER's VL_EUMIN input. These engineering limits will be conveyed to the CONTROLLER's VL_EUMAX_OUT and VL_EUMIN_OUT outputs, respectively. As configured, OUTPUT's upper limit (UHRANGE) and the OUTPUT's lower limit (ULRANGE) match the CONTROLLER's VL_EUMAX and VL_EUMIN, respectively

Configure the following parameters (which are marked for IEA) for your application:

- All tuning parameters (PG, TI, TD and DG)
- All high/low alarm and absolute deviation limits (H_LIM, L_LIM, AD_LIM)
- Valve scaling units (VL_EUMIN, VL_EUMAX)
- Valve "clamping" limits (VLH, VLL). Note that VLH cannot exceed 3% of span above VL_EUMAX and VLL cannot exceed 3% of span below VL_EUMIN.
- Setpoint limits (SPH, SPL)

Faults

Channel faults are detected by the input and output drivers and are assigned to input QBAD of the CONTROLLER block. An alarm message will then be posted if QUAL_EN is enabled.

See also:

SINGLE_LP_SS_AFB

3.14 VALVE_A_ALM_AT

This template can be used to implement standard valve open/close and device alarming functions. The VALVE block is an encapsulation of the VALVE_A block with a 4-element discrete alarm block. The VALVE block has inputs OP_VER, CL_VER, A_OPEN, A_CLSE, and INT0-INT10 for the open verify, close verify, auto open command, auto close command, and ten separate interlocks, and parameters OUT and FAIL for the output, and failure status of the valve, respectively. Alarms are provided for the following conditions:

- Failure to open (or loss of open verify)
- Failure to close (or loss of close verify)
- Uncommanded close/open or rejected command (due to interlock)
- Failed limit switch (Open and close verify inputs are set.)

The latched fail state of the VALVE_A block is unlatched automatically when the alarm is acknowledged. The valve circuit contains, apart from the valve block, two digital input drivers and one digital output driver.

Interconnections

The signal for the feedback (open/closed) must be connected to the VALUE inputs of blocks FB_OPEN and FB_CLOSED via symbolic addressing. The VALUE output of the OUTPUT block must be connected via symbolic addressing to the signal for the valve to be controlled.

Simulation

If there is no feedback, the simulation can be switched on at driver blocks FB_OPEN and FB_CLOSED (SIM_ON = 1). A connection is provided from the VALVE output OUT to a TIMER_P delay block (SIMULATION) IO input. The SIMULATION QO output is connected to the input SIM_I of block FB_OPEN. It is also connected to input SIM_I of block FB_CLOSED after being inverted. Simulation of valve limit switches is accomplished by delaying VALVE block's OUT signal. Note than the TIME0 parameter should be less than VALVE's OPEN_T and CLOS_T parameters.

Parameter Assignment

Configure the following parameters for your application:

- To set the VALVE block’s maximum allowed time to open and close, set the parameters OPEN_T and CLOS_T of the VALVE block (which are marked for IEA) individually.
- Interlocks and Interlock descriptions
- Permit and Permit description
- Fail-safe state of the valve (FAIL_OPEN)

Faults

Channel faults are detected by the input and output drivers.

See also:

VALVE_A_ALM_AFB

4 Message class

Classification of messages in accordance with their cause. The following message classes are used in the SIMATIC process control system:

- Process signals which are triggered when process-specific monitoring values (for example: alarms, interrupts, upper/lower tolerance, general process signals) are reached or exceeded.
- Control system messages which are output by the control system (system messages), the I/O units (errors in the field) or for preventive maintenance.
- Requests for operator input which, in the case of certain operation sequences, draw the operator's attention to the necessity of an operator intervention (for example, request to acknowledge a stepping operation manually in order to enable transition) or operator input lists.

Table of possible message classes and their meanings:

Message class	Meaning
AH	Alarm high (high high alarm)
WH	Warning high (high alarm)
WL	Warning low (low alarm)
AL	Alarm low (low low alarm)
TH	Tolerance high
TL	Tolerance low
F	Process error (field)
S	Control system message (system)
S*	OS control system message (fault)
M	Preventive maintenance
PM	Process message
-	Operation message
OR	Operator request
OM*1)	Operation message

*1) If the block is used for operation messages, the inputs I_1, ... have to be supplied with pulses. Assignment of the static value 1 would lead to multiple messages.

5 Differences between APACS+ and PCS 7

5.1 General Differences

Softlist Parameters

There are no softlist parameters in function blocks created in PCS 7. In general, these values have become In/Out parameters. In some specific cases, they are inputs.

Time Parameters

Time parameters have a different range of values. The maximum values are as follows:

APACS+ maximum time	49d17h2m47s295ms
PCS 7/APACS+ ES Libraries maximum time	24d20h31m23s647ms

Relationship Between Blocks and Charts

In APACS+, Derived Function Blocks can be opened in *4-mation* to give the user access to the standard function blocks that form the DFB. During runtime, the values of the component FBs can be viewed and modified. The DFBs also have faceplates to allow the values to be viewed and modified from an HMI. Source charts created in the CFC editor of the PCS 7 ES look very similar to DFBs opened in *4-mation*. However, the source charts are not open blocks; they are compiled to create function blocks. Both charts and the FBs created from them can be placed in CFCs, but only charts can be opened. On the other hand, faceplates can only be associated with function blocks, not charts.

Restart Logic Explanation

APACS+ uses local variables, Input/output nubs, and softlist parameters to define the operational state of a control algorithm. In APACS+, the initial value of local variables is used to automatically force the given state of a control object. For example, the control object Single_Loop can be forced to Auto on a cold start by setting the initial value of the local variable AUTO to TRUE. Within APACS+, the only way in which the initial value of AUTO can be changed is from the engineering console using *4-mation* to edit this value.

PCS 7 combines the functions of local variables, Input/output nubs, and softlist parameters into block parameters. Block parameters can be viewed as physical Input/output nubs on the block or by selecting to view the block parameters. You might say that all parameters are now "softlist" parameters that can also be viewed as block nubs on a per instance basis by the user. Referring back to the example of Single_Loop operating within PCS 7, the user can define the initial state of the parameter AUTO. This value will be used the first time the configuration is downloaded to the controller. A feature exists within PCS 7 that allows the user to read back parameters such as AUTO. Once this is done, the initially configured value will be overwritten. For example:

1. The original configuration of AUTO was defined as TRUE.
2. The configuration is then downloaded to the controller. This loop starts in Auto.
3. The operator changes the controller to Manual (AUTO=FALSE).
4. The user performs a read back or upload.
5. The user then downloads the configuration to the controller again.
6. This loop will now start in Manual and not Auto.

The addition of Restart Logic in the APACS+ Library blocks allows the user to define a restart state independent of the user performing a readback or upload of the controller configuration. See the help for each FB for an explanation of the restart parameters provided. See the section on Read Back (S7) for additional information on this PCS 7 functionality.

Read Back (S7)

You can save (read back) the parameters of all active CFC charts of the chart folder and changes to AS data (e.g. limit values or control parameters modified by the operator at an OS with PCS 7 OS).

You request to read back parameters via the **Chart > Read Back...** menu command.

When you select the "Online program" option in the "Read Back" dialog box to define the source file, the program of the CPU (online block folder) will be read back to the chart folder, including actual parameters.

Another read back method is to set the "Offline program" option in the "Read Back" dialog box to define the source file, after you have read back those data from the CPU in the online view via SIMATIC Manager (by means of the **PLC > Upload** menu command). Data will now be fetched from the offline block folder and written to the chart folder.

In either case, the dialog lets you choose to read back **all** parameters of the block inputs or only the parameters assigned the **operator control and monitoring** (S7_m_c:='true') system attribute.

In a PCS 7 project, only the "Parameters controllable in WinCC" option is useful and thus set for read back actions by default.

The following rules apply, irrespective of the scope set for the read back function:

- Only in_out and input parameters can be read back (not output parameters).
- The following data types are ignored when reading back data:

  ANY, ARRAY, POINTER, UDT.
- Inputs with write access by SFC chart are not read back.
- Certain blocks can be excluded from the read back operation (e.g. BATCH blocks). In this case, the block type is assigned the S7_read_back := 'false' attribute. This attribute cannot be modified in block instances.
- **Note on H CPUs:**

If the H CPU is operated in stand-alone mode (e.g. due to CPU failure) and a CPU fail-over has taken place, the user is prompted to select the required CPU after an online access request. This dialog does not appear if the CPU is operated in redundant mode.

- **Note on the download of HW Config data:**

Please note that the CPU will write data from load memory to work memory after you have downloaded HW Config data. Changes effected in debug mode or via operator control and monitoring only exist in work memory. Due to the transfer of data from load memory, all data in work memory will be lost, since its initial values will be overwritten by data from load memory. In this case you can create a backup of operator control data by means of read back, before you perform a download of HW Config data. You can then include these backup data in a program download to the CPU (in STOP mode).

The following applies for setting the scope of the "Parameters controllable in WinCC" function:

- Values of FC inputs will be ignored.

The following applies for setting the scope of "All parameters" function:

- FC inputs interconnected to chart I/Os will be read back. This also applies to chart I/Os with multiple interconnections to block inputs (FB and FC).

After read back:

- The full program is compiled automatically in order to ensure consistency. This does not, however require a download of the full program, i.e. you can also perform a delta download when the CPU is in RUN mode.
- A log file will be generated during read back and displayed after completion. You can always view this log via the **Options > Logs > Tab: Read Back** menu command. The log entries indicate
- I/Os read back and modified
- Where problems occurred

Batch Parameters

The SIMATIC Batch parameters are:

BA_EN	Batch enabled
OCCUPIED	Occupied by batch
BA_NA	Batch name
BA_ID	Batch ID number
STEP_NO	Batch step number

The batch parameters have been added as inputs to most of the control blocks for support of batch operations. A future Batch View faceplate or a customized faceplate may display the values in the HMI corresponding to these parameters.

The current values of BA_NA, BA_ID and STEP_NO are available for transmitting in alarm messages to the HMI, so that these values may be displayed and logged.

DecimalPlaces

Input parameters DecimalPlacesPV, DecimalPlacesOUT, etc have been added to loop control blocks to allow specifying the number of decimal places to display on the block's faceplate in the HMI. The range is 0 to 6, with a default value of 2.

ConfirmChange

The ConfirmChange input flag is added to most of the library blocks, to control operator changes in the HMI. The default value is FALSE, but this may be re-configured for any desired block instance.

If ConfirmChange is FALSE (default):

- Operator changes are "one-step", no further confirmation is required.
- all choices that are two-state, such as Auto/Manual, will immediately toggle to the other state.
- Increase/Decrease buttons will be visible and may be used by operator for quick changes of analog values, when appropriate (e.g. changing SP when in Auto mode).

If ConfirmChange is TRUE:

- Operator changes are "two-step", further confirmation is required. Usually an "input panel" will pop up, where operator may select a value and click OK.
- all choices including two-state choices require confirmation.
- Increase/Decrease buttons will be invisible.

Note that in both cases all operator changes are sent to the Operation List log.

Operator or Program control of important loop variables

Often there is a need for a program, for example a Batch program, to take control of one or more important loop variables such as SP, OUT, Auto/Manual, etc. At these times the operator in the HMI should not be able to control the variables in order to avoid conflict.

In APACS programs, the SET_VAL block could be used by a program to write to a variable and then return control of the same variable to the operator.

Instead, the New ES Library uses the PCS 7 technique of selecting between "Link" (i.e. a linked connection to a block in the program) and "Operator", the operator in the HMI. New parameters beginning with "LIOP_..." can be seen for this purpose. For example, the switch between Auto and Manual modes of operation can be made either by the Program or by the Operator, as follows:

- LIOP_MAN_SEL = TRUE: The connected value at the AUT_L ("link ") block input will determine the Auto/Manual mode. Operator input is ignored.
- LIOP_MAN_SEL = FALSE: The operator input from the HMI will determine the Auto/Manual mode. The value configured at the AUT_L block input is ignored.

In all cases the correct current value of the variable is shown to the operator in the faceplate.

Process Object View

Many of the configurable parameters of the library have been enabled for viewing and editing with the SIMATIC Process Object View tool in SIMATIC Manager, accessible under the "View" menu. This gives a convenient method for bulk programming of selected values in the program.

5.2 Alarm Differences

The philosophy of acknowledgement of alarms that are disabled or out-of-service differs between APACS+ and PCS 7. In APACS+, when an alarm is disabled or alarms are OOS, all affected current alarms are set FALSE and automatically acknowledged in the controller, but are still shown as active in the HMI. In PCS 7, alarms are acknowledged in the HMI and reported to the controller. When an alarm is disabled or alarms are OOS, all affected current alarms are set FALSE in the controller, but they are not automatically acknowledged in order to avoid a difference between the controller state and the state of the HMI. In the HMI, any affected alarm that was previously acknowledged will be removed from the incoming alarm list. Any affected alarm that was not previously acknowledged will show a change of status to indicate that the condition is gone. New alarms will not be posted in either PCS 7 or APACS+ if they are disabled.

5.3 APACS+ versus PCS 7 Controller Algorithm Comparison

The PID controller algorithms used by the APACS+ and the PCS 7 control systems are not identical. The major differences between the controller algorithms are the following:

- The APACS+ controller is based on the series/interacting transfer function while the PCS 7 controller is based on the parallel/non-interacting transfer function.
- The APACS+ controller implements the integral action using reset feedback whereas the PCS 7 controller implements the integral action directly to the gain*error.
- Derivative filtering in the APACS+ controller will always vary in direct proportion to the derivative time whereas the PCS 7 controller contains a separate derivative filter parameter that must be independently adjusted in relation to the derivative time.
- The time units of the integral and derivative components are in minutes for the APACS+ controller while they are in seconds for the PCS 7 controller.

APACS+ Algorithm

The transfer function used in the APACS+ controller (plus the Model 352 et. al.) is shown below:

$$OUT = PG\left(1+\frac{1}{TIs}\right)\left[\left(\frac{TDs}{\frac{TD}{DG}s+1}+1\right)PV - SP\right]$$

where: PG is proportional gain, TI is integral time, TD is derivative time, and DG is derivative gain.

In the APACS+ controller, the proportional gain value is always positive with the controller action determined by the state of the DIR_ACT (Direct Action) softlist parameter. With DIR_ACT = FALSE (the default), the controller acts as a reverse action controller.

The integral action is implemented using external reset feedback, which provides many benefits such as:

- the ability to initialize PID for bumpless transfer from Man to Auto without SP tracking
- the capability to implement numerous reset windup prevention strategies in cascade and override control
- a simple method of implementing deadtime compensation using complementary feedback

Integral action can be removed by setting TI = 4000.0.

The derivative action is always applied only to the PV, not to the error (PV – SP). Derivative action can be removed by setting TD = 0.0. Derivative filtering is

implemented as a function of both the derivative time and the derivative gain. The benefit of this is that since derivative filtering is always directly proportional to the derivative time for a given derivative gain, satisfactory derivative response is obtained over a wide range of derivative time values without requiring an adjustment to the derivative gain value. The derivative gain can be adjusted to provide more or less filtering, if desired, but the default value of 10.0 is usually sufficient for most applications.

PCS 7 Algorithm

The default transfer function used in the PCS 7 controller is shown below:

$$LMN = GAIN\left(1 + \frac{1}{TNs} + \frac{TVs}{1 + TM_LAGs}\right)(SP - PV_IN)$$

where: LMN is the controller output, TN is integral time, TV is derivative time, and TM_LAG is derivative filter time.

In the PCS 7 controller, the sign of the GAIN parameter determines the controller action. With a positive GAIN, the controller acts as a reverse action controller. Proportional action can be removed by setting (Proportional Select) P_SEL = 0. By optionally setting (Proportional Action in Feedback Path Select) PFDB_SEL = TRUE, the proportional action can be directly applied to the process variable, PV_IN, instead of the error.

The integral action is always applied to the gain*error and can be removed by setting TN = 0 or can be reversed by setting TN < 0. The controller does have a manipulated variable feedback signal input, LMNR_IN, however it is used only by the PCS 7 OS, it is not used as a reset feedback signal for implementing the integral action.

The derivative action can be removed by setting TV =0 or can be reversed by setting TV < 0. By optionally setting (Derivative Action in Feedback Path Select) DFDB_SEL = TRUE, the derivative action can be directly applied to the process variable, PV_IN, instead of the error. In this controller implementation, the derivative filter, TM_LAG, is completely independent of the derivative time, TV. The significance of this is that the derivative filter component is essentially a fourth tuning parameter that must be manually set separately and in relation to the derivative time; otherwise the derivative action will become too aggressive as the derivative time is increased. Normally, TM_LAG should be set to approximately (TV / 10) which effectively yields a derivative gain of 10.

Which controller algorithm to use?

The APACS+ controller algorithm is most similar to an analog controller and is reasonably tuned using the Ziegler-Nichols tuning method. The reset feedback implementation plays a key role in most of the basic and advanced control strategies. The PCS 7 controller algorithm is based on the theoretical "ideal" equation and provides many configuration options. This flexibility, however, requires more effort to set up.

As mentioned above, the implementation of derivative filtering is different. The APACS+ controller essentially makes derivative filtering "transparent" by adjusting the filtering in direct relation to the derivative time. The PCS 7 controller, however, requires manual intervention to maintain properly adjusted derivative filtering and someone not familiar with derivative filtering could become convinced that the PCS 7 derivative action does not work and quickly abandon its use.

Another distinction between the two algorithms is how the output behaves on changes to setpoint:

APACS+ has essentially one equation configuration:

- derivative action only on PV, yielding proportional and integral actions on SP changes

However, PCS 7 has a number of ways its equation can be configured, for example:

- derivative action only on PV, yielding proportional and integral actions on SP changes (same as APACS+)
- proportional action only on PV, yielding integral and derivative actions on SP changes
- proportional and derivative actions on PV, yielding only integral action on SP changes

In summary, there are advantages and disadvantages to both controller algorithm implementations. The decision of which controller to use is ultimately a matter of familiarity or personal preference.

6 Customizing the Library

These instructions describe customization of the source charts and files, including compilation to create new block types and modifcation of the graphical elements.

The following procedures are detailed below:

- Create new block types
- Modify graphical elements
- Test the changes

Create new block types	
Procedure	**Example**
1. From SIMATIC Manager, copy the Library.	1. From the SIMATIC Manager main menu, select **File > Open**. The **Open Project** dialog box appears. Click the **Libraries** tab, and select **APACS Library**. From the main menu, select **File > Save As New APACS Library**.
2. Open the Plant View of the Library.	2. From the main menu, select **View > Plant View**.
3. Copy the source chart to be modified.	3. In the **Source Charts\Control hierarchy** folder, copy and paste **SINGLE_LOOP**. Rename the copy to **NewSINGLE_LOOP**.
4. Make the necessary modifications.	4. Add Message Error and Message Status outputs to Type. a. Double-click **NewSINGLE_LOOP** to open it. b. Select the **ALARM** block on the chart. c. Right-click to edit the **Object Properties**. d. From the Properties dialog box, click the **Inputs/Outputs** tab. e. Uncheck the **Invisible Check Box** for the following Names: • MSG1_ERR • MSG1_STAT • MSG2_ERR • MSG2_STAT f. Click **OK**. g. From the main menu, select **View > Chart Inputs/Outputs**. The Chart I/O window for the NewSINGLE_LOOP appears. Click the **OUT** folder object. In the right-hand pane, scroll to the bottom. h. Click the **ALARM** block in the chart window. Select each of the newly visible parameters (MSG1_ERR...), and drag/drop to the last line of the Chart I/O OUT list window. i. For **each** of the newly inserted parameters, right-click the parameter object in the Chart I/O list. Select **Object Properties,** and click the **Attributes** tab. If not present, add (using the drop-down box) the attribute **S7_m_c**. Set the value for S7_m_c to **true**. Note that the new attribute will remain red until you select ok. j. Click **OK**.
5. Compile as a block type.	5. From the main menu, select **Chart > Compile > Chart as block type**.

6. Assign a new number, name and comment.	6. a. On the General tab, change Symbolic Name to **NewSLOOP**, change the FB Number to **650**. b. On the Attributes tab, change S7_m_c value to **true**.
7. Complete the compile.	7. Click **OK.**
	8. Close the window.
Modify Graphical Elements	
Note: All changes must be made to files in the siemens\wincc\options\pdl\faceplatedesigner_v6 directory for the changed files to be automatically copied to subsequent projects. If the changes are made in a project directory, they will only be effective in that project.	
Procedure	**Example**
1. Start Graphics Designer from an existing project in PCS 7 OS.	1. Double-click on **Graphics Designer** from PCS 7 OS, WinCC Explorer.
2. Update the block icon for the new type.	2. From Graphics Designer, open **siemens\wincc\options\pdl\faceplatedesigner_v6\ @PCS 7 Typicals.pdl**
3. Copy desired symbol (block icon).	3. Copy and paste the **Single_Loop_AFB** symbol to an empty space on the graphic.
4. Edit properties of the new symbol.	4. a. Right-click to edit properties of the new symbol. In the **Customized Object** property, change **Object Name** to **@NewSloop/1** by double-clicking on **Object Name** in the **Attribute** column. b. In UserDefined3 property, edit Type to **@NewSloop/1**, and edit Servername to **@NewSloop/1**. c. Click **OK**. d. Close the **Object Properties** window.
5. Save the file.	5. Save the @PCS 7 Typicals.pdl file.
6. From the siemens\wincc\options\pdl\faceplatedesigner_v6 directory in Windows Explorer:	6. From Windows Explorer, change the directory to: **siemens\wincc\options\pdl\faceplatedesigner_v6**

a. Copy @PG_originalname.PDL and save as @PG_newname.pdl b. Copy @PG_originalname_overview.PDL and save as @PG_newname_overview.pdl c.Copy@PG_originalname_standard.PDL and save as @PG_newname_standard.pdl d. Copy @PG_originalname_viewlist.PDL and save as @PG_newname_viewlist.pdl e. Copy @PL_originalname_overview.PDL and save as @PL_newname_overview.pdl	a. Copy @PG_Single_Loop_AFB.PDL and save as **@PG_NewSloop.pdl** b. Copy @PG_Single_Loop_AFB_overview.PDL and save as **@PG_NewSloop_overview.pdl** c. Copy @PG_Single_Loop_AFB_standard.PDL and save as **@PG_NewSloop_standard.pdl** d. Copy @PG_Single_Loop_AFB_viewlist.PDL and save as **@PG_NewSloop_viewlist.pdl** e. Copy @PL_Single_Loop_AFB.PDL and save as **@PL_NewSloop.pdl**
7. Update Faceplate Overview.	7. a. Double-click **@PL_NewSloop.pdl** to open it. b. From the main menu, select **View > Properties**. The **Object Properties** dialog box opens. Use the drop-down menu to select the **Overview Window** object. Click the **Properties** tab, in **Miscellaneous** property, change Picture Name to **@PG_NewSloop_Overview**. c. Use the drop-down menu to select the **Block Type** object. Click the **Properties** tab, in Output/Input property, change Output Value to **NewSloop**. d. Use the drop-down menu to select the **View1** object. In Miscellaneous property, change Picture Name to **@PG_NewSloop_Standard.pdl**. e. Save the file.
8. Update Faceplate Header.	8. a. Open **@PG_NewSloop.pdl**. b. From the main menu, select **View > Properties**. The Object Properties window appears. Use the drop-down menu to select the **@faceplate** object, in Texts property, change First View to **@PG_NewSloop_Standard**. c. Save the file.

9. Update the Standard View.	9. a. Open **@PG_NewSloop_Standard.pdl**. b. From the **Object Palette** window, in the **Standard Objects** folder, drag a **Static Text** object to any open space in the graphic. Draw a text box, and type **Error_1** in the new field. c. From the **Object Palette** window, in the Smart Objects folder, drag an **I/O Field** onto the graphic next to Error1, and draw a text box. Click **OK**. d. Right-click the new I/O Field. The **Object Properties** dialog box appears. Choose the Output/Input property, select Output Value, and change the Dynamic to **.MSG_ERR** e. Change Output Format to **99** f. Copy both new fields, and paste them into any open space in the graphic. g. Change the text in the copied Static Text object to **Status1**. h. Right-click on the new I/O Field. The **Object Properties** dialog box appears. Choose the Output/Input property, select Output Value, and change the Dynamic to **.MSG_STAT** i. Save the file.

Test the changes
Procedure
1. Create a new single project using project wizard. Include a PCS 7 OS. Configure the hardware and SIMATIC PC station according to your system.
2. From Plant View, open the CFC that was created by default.
3. Drag the new block from the catalog to the CFC editing area.
4. Compile the Chart.
5. Download the program.
6. In Plant View, edit the properties of the picture at the hierarchy level of the chart.
7. On the Block Symbols tab, check the box "**Derive the block symbols from the technological hierarchy**". Click **OK**.
8. In SIMATIC Manager, compile the OS. Check the box "**Create\update block icons**"
9. Open the OS for the project.
10. Start the runtime OS.
11. In the WinCC runtime, navigate to the picture that was updated with the block icons.
12. Verify that your icon appears on the plant picture.
13. Click on the icon to open the faceplate.
14. Verify the new fields are on the faceplate.
15. Open the Loop display.
16. Verify the correct name of the type which appears below the title bar.

7 Performance Data

The column headings for the performance data table have the following meanings:

Block (Type Name)

The symbolic identifier in the symbol table of the library for the respective FB or FC.

FB No.

Block number

Typical run time

The time which the CPU needs to process the corresponding block program under normal circumstances (for example, for a driver this is the execution time in the watchdog interrupt OB, without message generation in the event of channel errors).

The following table shows the block data for blocks in a CPU S7 416-3 DP 6ES7 416-3XL00-0AB0. With other CPUs the run time depends on their performance.

Block length

Memory requirements of the program code, once for every block type.

Instance data length

Memory requirement of an instance DB.

Temporary memory

The local-data memory required in a priority class when the block is called. This is limited to the CPU specifically. When it is exceeded, a CPU STOP is caused. You have to check it in the CPU configuration and, if necessary, distribute it amongst the priority classes (Obs) to meet the real requirements.

Multiple instance block

The specified blocks are used by the technological block and must exist in the user program. They can be found in the same library.

Block (Type Name)	FB No.	Typical run time (ms)* CPU 416-3	Block length (bytes)	Instance data length (bytes)	Temporary memory (bytes)	Multiple Instance block
AM_AFB	FB 500	0.045	540	90	48	
PID_AFB	FB 501	0.403	1994	190	94	
SETPNT_AFB	FB 502	0.075	1854	134	78	
AALM_AFB	FB 503	0.116	5296	550	146	2xSFB35
DALM_AFB	FB 504	0.154	2096	290	26	SFB35
MOTOR_A_AFB	FB 505	0.198	1598	746	50	FB504
VALVE_A_AFB	FB 506	0.199	1938	712	50	FB504
ANALOG_ALM_AFB	FB 600	0.160	1134	758	2	FB503
BLK_VLV_1OUT_AFB	FB 601	0.196	856	368	50	2xFB5, FB504
BLK_VLV_2OUT_AFB	FB 602	0.197	1104	366	50	2xFB5, FB504
CASCADE_AFB	FB 603	1.359	5308	2222	48	FB500, 2xFB501, 2xFB502, 2xFB503
DISCRETE_ALM_AFB	FB 604	0.144	526	368	2	FB504
EXT_SET_AFB	FB 605	0.732	3116	1180	48	FB500, FB501, FB502, FB503
MOTOR_1OUT_AFB	FB 606	0.199	866	370	54	2xFB5, FB504
MOTOR_2OUT_AFB	FB 607	0.199	1068	370	54	2xFB5, FB504
PRIMARY_AFB	FB 608	0.755	3208	1166	44	FB500, FB501, FB502, FB503
RATIO_SET_AFB	FB 609	0.843	4268	1360	48	FB500, FB501, 2xFB502, FB503
SECONDARY_AFB	FB 610	0.732	3136	1178	48	FB500, FB501, FB502, FB503
SINGLE_LOOP_AFB	FB 611	0.730	3044	1178	48	FB500, FB501, FB502, FB503
SINGLE_LP_SS_AFB	FB 612	0.732	3118	1182	48	FB500, FB501, FB502, FB503
MOTOR_A_ALM_AFB	FB 613	1.067	6442	1530	56	FB504, FB505, FB9
VALVE_A_ALM_AFB	FB 614	1.049	6280	1464	56	FB504, FB506, FB9

*Note: The typical run times shown above were measured when alarms were not occurring. For blocks with alarming capability, execution time will increase when the block has an unacknowledged alarm. For blocks with analog alarms, approximately .140 milliseconds will be added to the average time shown in the table for each message containing at least one unacknowledged alarm. For blocks with discrete alarms, approximately .125 milliseconds will be added to the average time shown in the table for each message containing at least one unacknowledged alarm.